

# Dokumentumszerkesztés $\LaTeX$ kurzus

Telek Miklós

BME, Hálózati Rendszerek és Szolgáltatások Tanszék

(Jeney Gábor és Horváth Máté fóliái alapján)

2024. február 14.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

A  $\LaTeX$  használatának alapjai

A szöveg begépelése

A dokumentum struktúrálása

Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

Táblázatok, ábrák

Táblázatok

Képek, ábrák

Matematikai képletek I

Matematikai képletek II

Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

Dobozok

Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

## Dokumentumszerkesztés – 2 kredit

**FIGYELEM!** A kurzus kódja meghatározza a szövegtördelő alkalmazást:

- ▶ Lx –  $\LaTeX$
- ▶ Wx – M\$ Word

Érdemes figyelni továbbá a kurzus napját (kedd, szerda, csütörtök), mindegyik kurzus 12:15-kor indul.

Mindenki ellenőrizze, hogy a megfelelő kurzusra jelentkezett-e.

Ez itt a  $\LaTeX$  kurzus.

Telek Miklós, I.L.117, Tel: 2084, [telek@hit.bme.hu](mailto:telek@hit.bme.hu).

Tegezés:

→ Ami jól esik (mivel évről évre nő a korkülönbség)!

← Általában tegeztek, néha elhibázom (hasonló okból)!

# Időbeosztás

Az órák 12:15-kor kezdődnek és 13:45-kor érnek véget (nincs szünet).

Tipikus időbeosztás:

- ▶ 12:15-13:00 Új ismeretek,
- ▶ 13:00-13:45 Gyakorlás oktatói segítséggel.

Gyakorlás: A félév során 9 alkalommal kb 13:00-tól 13:45-ig kilépő feladatot lehet megoldani, amelyeket az óra vége előtt lehet feltölteni a <http://moodle.hit.bme.hu/> oldalon.

Ezekből a feladatokból legjobb 6 eredménye játszik szerepet a végső jegyben.

Csak a nem hiányzók kilépő feladatát értékelem.

# Kilépő feladatok:

- ▶ **Motivált gyakorlás.**
- ▶ Elakadások esetén **oktatói segítség.**
- ▶ Segít megérteni az anyag elsajátításának szintjét.
- ▶ Előkerülnek azok a hibák amiket a nagy feladatban el kell kerülni.
- ▶ Segít észlelni és leszokni a berögzült szerkesztési hibákról.

*Az első kilépőknél a gyarkolat hiánya miatt nem mindenkinek elég az idő.*

Ne izguljatok e miatt. A későbbi kilépők általában sokkal gyorsabban elkészülnek. (Ez is a gyakorlás célja.)

# A tárgy adminisztrációja

A tárgy adminisztrációját a <http://moodle.hit.bme.hu/> oldalon végezzük.

Felhasználó név: NEPTUN kód (kis betűkkel)

Jelszó: jelszó emlékeztető a NEPTUN-ban megadott e-mail címre.

Az oldalon a tárggyal kapcsolatos bejegyzések folyamatosan ellenőrizhetők.

Ha valaki hibás adatot (pl. hiányzó kilépő eredmény) lát a moodle oldalon kérem azonnal jelezze.

A (kilépő és nagy) feladatok beadása is ezen az oldalon történik.

**A határidő letelte után a rendszer nem engedi meg a feladatok beadását.**

Időben kezd el feltölteni a megoldásodat!!

Aláírást az kaphat, aki az órák kevesebb mint 30 %-áról hiányzik (maximum 4 óra), és többin ott van.

A kilépő feladatok elkészítése opcionális, tehát nem kötelező, viszont a végső jegy harmadát ezek megoldása adja.

A félév végi jegyet döntő részben (2/3) a nagy házi feladatra adom. A feladatot a 10. hét környékén osztom ki, és a 12. hét környékén kell beadni a moodle rendszerben. **A határidő letelte után a rendszer nem engedi meg a feladatok beadását.**

Az utolsó órát a nagy házi feladat értékelésére fordítjuk, amelyen tipikusan a beadott dolgozathoz kapcsolódó kisebb szerkesztési módosítást kell elvégezni.



## Elektronikus jegyzetek

(<http://webspn.hit.bme.hu/~telek/latex>):

- ▶ Órák fóliái (ez a fólia sor),
- ▶ Alapvető tipográfiai ismeretek,
- ▶ Minta források a gyakorlatokhoz,
- ▶ Kapcsolódó linkek gyűjteménye.

További nyomtatott irodalmak (az első kettő tipográfiai, a másik kettő  $\text{\LaTeX}$ ):

- ▶ Virágvölgyi Péter, *A tipográfia mestersége*, Osiris, 2001.
- ▶ Gyurgyák János, *Szerkesztők és szerzők kézikönyve*, Osiris, 1996.
- ▶ Leslie Lamport,  *$\text{\LaTeX}$* , 2nd Ed., Addison-Wesley, 1994.
- ▶ Wettl, Mayer, Sudár,  *$\text{\LaTeX}$  kezdőknek és haladóknak*, Panem, 1998.

A tárgy azért indult, hogy a hallgatói dolgozatok (diploma, TDK, önlab beszámoló stb.) megfelelő minőségű elkészítését segítse. Ehhez szükséges beszélni

- ▶ az alapvető tipográfiai szabályokról,
- ▶ a tipikus számítógépes grafikus leíró eszközökről,
- ▶ a szövegtördelő programok tipikus szolgáltatásairól (pl. tartalomjegyzék).

A tárgy célja, hogy

- ▶ betekintést nyújtson azokba az alapvető tipográfiai szabályokba, melyek ismerete elvárható egy diplomás embertől,
- ▶ megtanítsa a **L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>**, vagy M\$ Word használatát, a beszámolók írására fókuszálva.

Gyakorolj az órán kívül is! Ahogyan programozni sem lehetett megtanulni csak az egyetemi órákon, úgy a  $\LaTeX$  sem elsajátítható órán kívüli gyakorlás nélkül.

Próbáld meg az összes tárgyból a beadandó feladataidat  $\LaTeX$  szövegszerkesztővel írni.

A  $\LaTeX$  nagyobb terjedelmű művek szerkesztésére igazán hatékony.

**A nagyobb struktúrák kezelését bemutató gyakorlatokhoz szerezz be egy legalább két szinten (fejezet, alfejezet) tagolt 5-8 oldalas szöveget.**

NEM lorem ipsum!

# A tárgy oktatásának módja (jelenléti)

Számítógépes tanórák keretében a egyetem valamelyik számítástechnikai laboratóriumában, heti két órában.

Az elsajátítást segíti, hogy a tanult ismeretek azonnal kipróbálhatóak a számítógépen.

Használd a teremben lévő gépeket, vagy hozd be a saját laptopodat, és folyamatosan próbáld ki az elhangzottakat az órákon.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

### A $\LaTeX$ születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

## Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat





- ▶ Elnevezés a görög  $\tau\acute{\epsilon}\chi\nu\eta$  (Technikai tudás, képesség az alkotásra) szóból
- ▶  $\TeX$  kiejtése: „teksz” „teh” (bár utal a text szóra is)
- ▶ Mi is a  $\TeX$ ?
  - ▶ egy betűszedő rendszer
  - ▶ része a METAFONT, ami betűkészlet létrehozására alkalmas (pl. a  $\TeX$ -é is ezzel készült)
- ▶ a  $\TeX$ verziószáma a  $\pi$ -hez, a METAFONT-é az  $e$ -hez konvergál
- ▶ az aktuális verziójuk 3.14159265 illetve 2.7182818 (2014 óta)
- ▶ nagyon stabil program: Knuth pénzdíjat ajánlott fel a megtalált hibákért (1 „hexadecimális dollárt” = \$2.56, amit évente duplázott)
- ▶ „A  $\TeX$  a nyomdászat assemblere.”





# A $\LaTeX$ születése

- ▶  $\TeX$ -ben több száz elemi parancs
- ▶ makrókat is lehet írni benne a dokumentumszerkesztés leegyszerűsítésére
- ▶ Makrócsomagok:
  - ▶ „plain  $\TeX$ ” by Knuth, alapkönyve a  $\TeX$ Book
  - ▶ *A.M.S.*- $\TeX$  by the American Mathematical Society
  - ▶  $\LaTeX$  by Leslie Lamport, alapkönyve a  $\LaTeX$



- ▶ Lamport a '80-as években kezdett dokumentumszerkesztője megírásába
- ▶ Az első hivatalos verzió a  $\LaTeX$  2.09
- ▶ 1989: Lamport visszavonul a fejlesztéstől
- ▶  $\LaTeX$ 3 munkacsoport: Frank Mittelbach, Chris Rowley és Rainer Schöpf
  - ▶ cél: újraírni a  $\LaTeX$ -et, hogy az egyszerűbb, egységesebb és széleskörűbb legyen
  - ▶ elhúzódo megvalósítás
  - ▶ a  $\TeX$ / $\LaTeX$  közösség tovább oszlott
- ▶ 1994: a munkacsoport kiadja az (előzetes)  $\LaTeX$ 2 $\epsilon$ -t
  - ▶ magába olvasztotta a legtöbb jó és elterjedt megoldást
  - ▶ jelenlegi hivatalos változat
- ▶ Mi is ezt használjuk a félév során.

# $\LaTeX$ vs. Word/OpenOffice

$\LaTeX$  (ami alatt a  $\LaTeX 2_{\epsilon}$ -t értjük mostantól):

**Word:**

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléi alapján)

**Nulladik gyakorlat**

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# L<sup>A</sup>T<sub>E</sub>X vs. Word/OpenOffice

**L<sup>A</sup>T<sub>E</sub>X** (ami alatt a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-t értjük mostantól):

- ▶ Globális szemléletű (ne foglalkozzunk/ne kelljen foglalkozni a részletekkel!)

**Word:**

- ▶ Lokális szemléletű (helyben gyorsan és hatékonyan lehet vele átformázni a szöveget, de nagyban nehéz ugyanezt csinálni)

# L<sup>A</sup>T<sub>E</sub>X vs. Word/OpenOffice

**L<sup>A</sup>T<sub>E</sub>X** (ami alatt a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-t értjük mostantól):

- ▶ Globális szemléletű (ne foglalkozzunk/ne kelljen foglalkozni a részletekkel!)
- ▶ Eleinte nehéz lehet megszokni, később viszont könnyebbnek érezzük a dokumentum átformálását (ha éppen szükséges)

**Word:**

- ▶ Lokális szemléletű (helyben gyorsan és hatékonyan lehet vele átformázni a szöveget, de nagyban nehéz ugyanezt csinálni)
- ▶ Felhasználóbarát felület → könnyedén kezelhető

# L<sup>A</sup>T<sub>E</sub>X vs. Word/OpenOffice

**L<sup>A</sup>T<sub>E</sub>X** (ami alatt a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-t értjük mostantól):

- ▶ Globális szemléletű (ne foglalkozzunk/ne kelljen foglalkozni a részletekkel!)
- ▶ Eleinte nehéz lehet megszokni, később viszont könnyebbnek érezzük a dokumentum átforgalmazását (ha éppen szükséges)
- ▶ Hosszú művekhez javasolt az ismerete

**Word:**

- ▶ Lokális szemléletű (helyben gyorsan és hatékonyan lehet vele átforgalmazni a szöveget, de nagyban nehéz ugyanezt csinálni)
- ▶ Felhasználóbarát felület → könnyedén kezelhető
- ▶ Nagy művek kinézetének egységes átalakítása embert próbáló feladat



# L<sup>A</sup>T<sub>E</sub>X vs. Word/OpenOffice

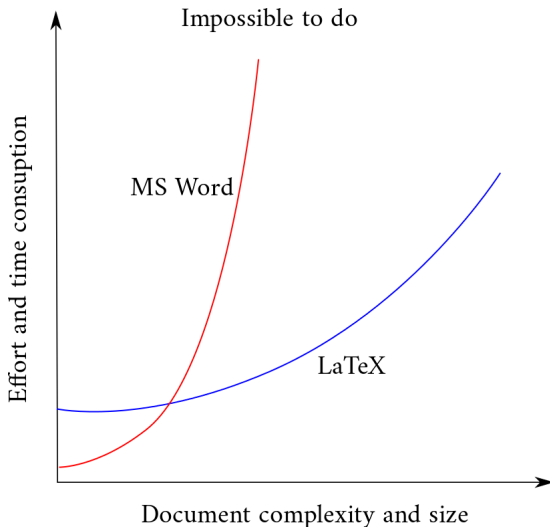
**L<sup>A</sup>T<sub>E</sub>X** (ami alatt a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-t értjük mostantól):

- ▶ Globális szemléletű (ne foglalkozzunk/ne kelljen foglalkozni a részletekkel!)
- ▶ Eleinte nehéz lehet megszokni, később viszont könnyebbnek érezzük a dokumentum átformálását (ha éppen szükséges)
- ▶ Hosszú művekhez javasolt az ismerete
- ▶ Egységes esztétikus megjelenést támogat

**Word:**

- ▶ Lokális szemléletű (helyben gyorsan és hatékonyan lehet vele átformázni a szöveget, de nagyban nehéz ugyanezt csinálni)
- ▶ Felhasználóbarát felület → könnyedén kezelhető
- ▶ Nagy művek kinézetének egységes átalakítása embert próbáló feladat
- ▶ Könnyű eseti stílusváltoztatást támogat.

# LaTeX vs. Word/OpenOffice



Hasonlítsunk össze az eredményt is! Az egyik illetve a másik.

# LaTeX vs. Word/OpenOffice

A különbség valójában az eltérő megközelítés eredménye:

WYSIWYM vs. WYSIWYG

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# LaTeX vs. Word/OpenOffice

A különbség valójában az eltérő megközelítés eredménye:

WYSIWYM vs. WYSIWYG

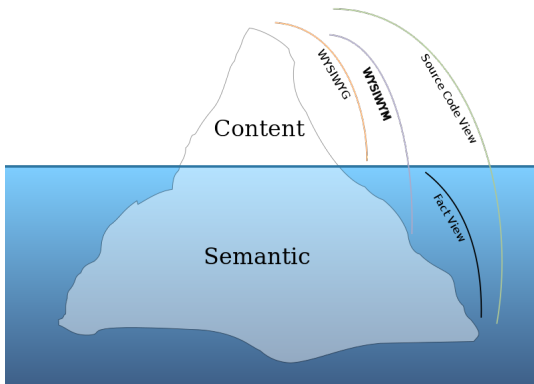
What You See Is What You **Mean / Get**

# LaTeX vs. Word/OpenOffice

A különbség valójában az eltérő megközelítés eredménye:

WYSIWYM vs. WYSIWYG

What You See Is What You **Mean / Get**



# Átjárás a paradigmák közt

- ▶ Windows alatt a ScientificWord és ScientificWorkPlace ([www.sciword.co.uk](http://www.sciword.co.uk)) a  $\LaTeX$  és a  $\TeX$  motorját használja a begépelte információ megjelenítéséhez, ám WYSIWYG megjelenítést biztosít
- ▶ a Word2TeX alkalmazás ([www.sciencesoftware.com/Word2TeX.asp](http://www.sciencesoftware.com/Word2TeX.asp)) az MS. Wordbe beépülve lehetővé teszi a  $\LaTeX$  formátumba mentést (ne számítsunk a saját munkához hasonló kimenetre...)
- ▶ a LyX ([www.lyx.org](http://www.lyx.org)), vagy  $\TeX$ macs ([www.texmacs.org/](http://www.texmacs.org/)) programok grafikus felületet biztosítanak a  $\LaTeX$  használatához

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

### $\LaTeX$ kimeneti formátumok

Vector vs. raszter

A  $\LaTeX$  használatának alapjai

A szöveg begépelése

A dokumentum struktúrálása

Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

Táblázatok, ábrák

Táblázatok

Képek, ábrák

Matematikai képletek I

Matematikai képletek II

Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

Dobozok

Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

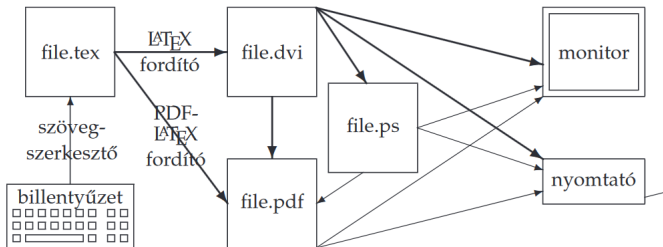
Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

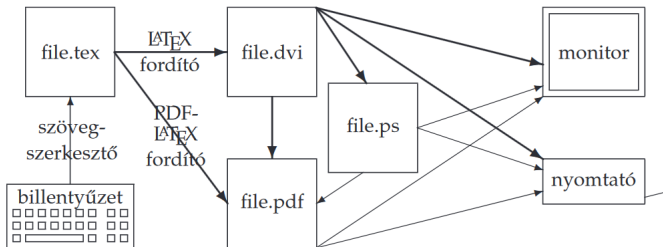
# A szerkesztés menete $\text{\LaTeX}$ -ben



Forrás:  $\text{\LaTeX}$  kézikönyv



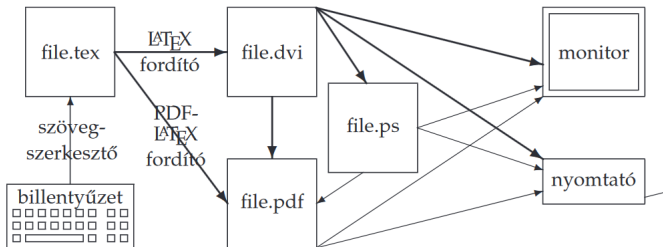
# A szerkesztés menete $\text{\LaTeX}$ -ben



Forrás:  $\text{\LaTeX}$  kézikönyv

- ▶ **DVI (DeVice Independent)** - Egy dokumentum vizuális tulajdonságait írja le (például font, margók, de NEM tartalmazza pl. a médiafájlokat, fontokat közvetlenül) → nem függ a megjelenítő eszköztől. Tipikusan egy másik program (DVI driver) bemenete, ami vizuálisan megjeleníthető alakra hozza.

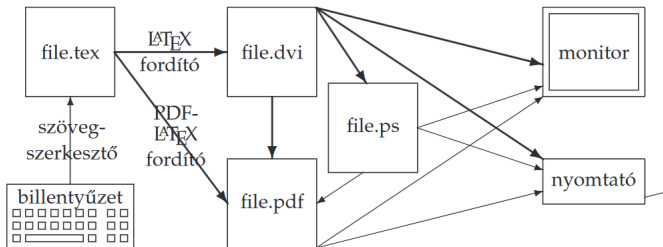
# A szerkesztés menete $\text{\LaTeX}$ -ben



Forrás:  $\text{\LaTeX}$  kézikönyv

- ▶ **DVI (DeVice Independent)** - Egy dokumentum vizuális tulajdonságait írja le (például font, margók, de NEM tartalmazza pl. a médiafájlokat, fontokat közvetlenül) → nem függ a megjelenítő eszköztől. Tipikusan egy másik program (DVI driver) bemenete, ami vizuálisan megjeleníthető alakra hozza.
- ▶ **PS (PostScript)** - Oldalleíró nyelv, amely a grafikai-tipográfiai elemek mindegyikét vektorgrafikus módon tárolja. ASCII formátumú, eszköz- és felbontásfüggetlen.

# A szerkesztés menete $\text{\LaTeX}$ -ben



Forrás:  $\text{\LaTeX}$  kézikönyv

- ▶ **DVI (DeVice Independent)** - Egy dokumentum vizuális tulajdonságait írja le (például font, margók, de NEM tartalmazza pl. a médiafájlokat, fontokat közvetlenül) → nem függ a megjelenítő eszköztől. Tipikusan egy másik program (DVI driver) bemenete, ami vizuálisan megjeleníthető alakra hozza.
- ▶ **PS (PostScript)** - Oldalleíró nyelv, amely a grafikai-tipográfiai elemek mindegyikét vektorgrafikus módon tárolja. ASCII formátumú, eszköz- és felbontásfüggetlen.
- ▶ **PDF (Portable Document Format)** - Az Adobe által kifejlesztett, dokumentumok tárolására alkalmas fájlformátum. Alkalmas szöveget, ábrát és képeket tartalmazó dokumentum leírására eszközfüggetlen és felbontásfüggetlen formában. Nyílt szabvány.

# A $\LaTeX$ folyamatosan fejlődik

$\LaTeX$  initial release: 1985

( $\TeX$ : 1978)

hardver (pl. grafikus felbontás), operációs rendszer, software

$\TeX$ ,  $\LaTeX$   $\rightarrow$   $\LaTeX$ 2 $\epsilon$

dvi, ps  $\rightarrow$  pdf

parancs sor  $\rightarrow$  integrált környezet

egyéni segéd csomagok  $\rightarrow$  integrált csomagok

bitmap  $\rightarrow$  vektor grafika

sajat gép  $\rightarrow$  online (Overleaf, ShareLaTeX, ...)

Ezt a fejlődést félévről félévre ebben a dokumentumban is megpróbáljuk követni, de az állítások helyessége időfüggő.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

### Vector vs. raszter

A  $\LaTeX$  használatának alapjai

A szöveg begépelése

A dokumentum struktúrálása

Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

Táblázatok, ábrák

Táblázatok

Képek, ábrák

Matematikai képletek I

Matematikai képletek II

Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

Dobozok

Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

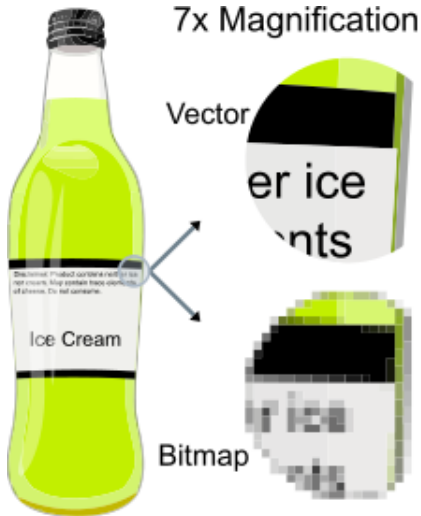
Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Vektorgrafika vs. rastergrafika



# Vektorgrafika vs. rastergrafika

## Vektorgrafika (SVG, EPS, PDF)

- ▶ geometriai primitíveket (rajzelemeket), pl. pontokat, egyeneseket, görbéket és sokszögeket használ képek leírására

## Rastergrafika (BMP, JPEG, PNG, APNG, GIF)

- ▶ a kép minden egyes pontjához egy érték van rendelve, mely a pont színéről vagy szürkeárnyalatáról ad felvilágosítást

# Vektorgrafika vs. rastergrafika

## Vektorgrafika (SVG, EPS, PDF)

- ▶ geometriai primitíveket (rajzelemeket), pl. pontokat, egyeneseket, görbéket és sokszögeket használ képek leírására
- ▶ a memóriaigény nem függ az objektum méreteitől

## Rastergrafika (BMP, JPEG, PNG, APNG, GIF)

- ▶ a kép minden egyes pontjához egy érték van rendelve, mely a pont színéről vagy szürkeárnyalatáról ad felvilágosítást
- ▶ a memóriaigény függ az objektum méreteitől



# Vektorgrafika vs. rastergrafika

## Vektorgrafika (SVG, EPS, PDF)

- ▶ geometriai primitíveket (rajzelemeket), pl. pontokat, egyeneseket, görbéket és sokszögeket használ képek leírására
- ▶ a memóriaigény nem függ az objektum méreteitől
- ▶ tetszőlegesen nagy kinagyítás sem torzítja el a képet

## Rastergrafika (BMP, JPEG, PNG, APNG, GIF)

- ▶ a kép minden egyes pontjához egy érték van rendelve, mely a pont színéről vagy szürkeárnyalatáról ad felvilágosítást
- ▶ a memóriaigény függ az objektum méreteitől
- ▶ a nagyítás torzítja a képet

# Vektorgrafika vs. rastergrafika

## Vektorgrafika (SVG, EPS, PDF)

- ▶ geometriai primitíveket (rajzelemeket), pl. pontokat, egyeneseket, görbéket és sokszögeket használ képek leírására
- ▶ a memóriaigény nem függ az objektum méreteitől
- ▶ tetszőlegesen nagy kinagyítás sem torzítja el a képet
- ▶ a vonalvastagságnak nem kell a nagyítással arányosan nőni

## Rastergrafika (BMP, JPEG, PNG, APNG, GIF)

- ▶ a kép minden egyes pontjához egy érték van rendelve, mely a pont színéről vagy szürkeárnyalatáról ad felvilágosítást
- ▶ a memóriaigény függ az objektum méreteitől
- ▶ a nagyítás torzítja a képet
- ▶ a vonalvastagság a nagyítással arányosan nő

# Vektorgrafika vs. rastergrafika

## Vektorgrafika (SVG, EPS, PDF)

- ▶ geometriai primitíveket (rajzelemeket), pl. pontokat, egyeneseket, görbéket és sokszögeket használ képek leírására
- ▶ a memóriaigény nem függ az objektum méreteitől
- ▶ tetszőlegesen nagy kinagyítás sem torzítja el a képet
- ▶ a vonalvastagságnak nem kell a nagyítással arányosan nőni
- ▶ az alakzatok méretei (paraméterek) tárolhatók és így később megváltoztathatók

## Rastergrafika (BMP, JPEG, PNG, APNG, GIF)

- ▶ a kép minden egyes pontjához egy érték van rendelve, mely a pont színéről vagy szürkeárnyalatáról ad felvilágosítást
- ▶ a memóriaigény függ az objektum méreteitől
- ▶ a nagyítás torzítja a képet
- ▶ a vonalvastagság a nagyítással arányosan nő
- ▶ egyszerű adatszerkezet; egyszerű algoritmus; gyors feldolgozás

# Vektorgrafika vs. rastergrafika

## Vektorgrafika (SVG, EPS, PDF)

- ▶ geometriai primitíveket (rajzelemeket), pl. pontokat, egyeneseket, görbéket és sokszögeket használ képek leírására
- ▶ a memóriaigény nem függ az objektum méreteitől
- ▶ tetszőlegesen nagy kinagyítás sem torzítja el a képet
- ▶ a vonalvastagságnak nem kell a nagyítással arányosan nőni
- ▶ az alakzatok méretei (paraméterek) tárolhatók és így később megváltoztathatók
- ▶ az adatok tárolhatók eszközfüggetlen egységekben → optimális rasterizálás

## Rastergrafika (BMP, JPEG, PNG, APNG, GIF)

- ▶ a kép minden egyes pontjához egy érték van rendelve, mely a pont színéről vagy szürkeárnyalatáról ad felvilágosítást
- ▶ a memóriaigény függ az objektum méreteitől
- ▶ a nagyítás torzítja a képet
- ▶ a vonalvastagság a nagyítással arányosan nő
- ▶ egyszerű adatszerkezet; egyszerű algoritmus; gyors feldolgozás
- ▶ az adatállomány nagy méretű; rögzített felbontás

# L<sup>A</sup>T<sub>E</sub>X forrás és a lefordított dokumentum

- ▶ Egy mai L<sup>A</sup>T<sub>E</sub>X szerkesztő környezetben (**ami a forrást mutatja nekünk**) az összes tipikus funkció könnyen elérhető. Így elegendő a szerkesztőt (pl. T<sub>E</sub>Xstudio) elindítanunk.
- ▶ A L<sup>A</sup>T<sub>E</sub>X szövegszerkesztőben egy gomb megnyomásával indítható a dokumentum
  - ▶ fordítása
  - ▶ a lefordított dokumentum megjelenítése
- ▶ Érdemes ismerni a parancssoros fordítás lépéseit is: (bonyolult feladatoknál, pl. index lista, jól jöhet)
  - ▶ `latex_file` → fordítás dvi-ba
  - ▶ `pdflatex_file` → fordítás pdf-be
- ▶ DVI fájlok megtekintéséhez
  - ▶ `yap.exe` (Win)
  - ▶ `xdvi` (Linux)

# Szoftver környezet a laborban

- ▶ Windows operációs rendszer,
- ▶ (elvileg teljes) T<sub>E</sub>XLive disztribúció ([www.tug.org/texlive/](http://www.tug.org/texlive/))
- ▶ szövegszerkesztéshez
  - ▶ T<sub>E</sub>Xstudio ([www.texstudio.org/](http://www.texstudio.org/))
- ▶ esetleg hiányzó L<sup>A</sup>T<sub>E</sub>X csomagokat órán kell letölteni (létrehozni) pl. a [www.ctan.org/](http://www.ctan.org/) oldalról
- ▶ PDF megjelenítő

## Bejelentkezés címtár azonosítóval

### Órai munka tárolása:

- ▶ *felhasználó azonosító* nevű könyvtárban és az alatt
- ▶ Ezeket hetente, néha ritkábban, törlik  $\implies$  saját célra másolat!
- ▶ **Szövegszerkesztőben szerkesztett adatok elveszhetnek** (pl. véletlen kikapcsolás esetén), illetve a kiugró feltöltéséhez is menteni kell az állományokat!

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Szoftver környezet az QF12-ben

- ▶ terminál indítása, browser indítása
- ▶ `cloud.bme.hu` és bejelentkezés EDU-ID azonosítással (akinek nincs EDU-ID azonosítója a `login.bme.hu` oldalon Neptun bejelentkezéssel éri el az azonosító számot, és jelszót is be kell állítani.)
- ▶ Belépés után „Smallville” kiválasztása
- ▶ ott: „virtual machines” new
- ▶ „Dokszerkesztés (Latex, Word) v1” start
- ▶ mikor a virtuális gép elindult „távoli asztal” (remote desktop) indítása
  - ▶ parancs sorból:  
`rdesktop vm.smallville.cloud.bme.hu:vmid -u cloud -p pwd -f`
  - ▶ vagy windows alkalmazás indítása.

Órai munka tárolása: internet

*Virtuális gépek megmaradásában nem szabad bízni.*



# Otthoni telepítés 1.

Ahogy programozni, úgy a  $\text{\LaTeX}$  használatát sem lehet megtanulni az egyetemi órákon otthoni gyakorlás nélkül.

- ▶ A  $\text{\LaTeX}$  ingyenes program, szabadon terjeszthető/telepíthető.
- ▶ Teljes működőképességéhez szükséges fájlokat disztribúciókban „szállítják”,

Windows esetén

- ▶ régebben elterjedtebb a Mik $\text{\TeX}$ : [www.miktex.org](http://www.miktex.org)
- ▶ ma már tipikusabb a  $\text{\TeX}$ live [www.tug.org/texlive](http://www.tug.org/texlive)
- ▶ Szövegszerkesztéshez:
  - ▶  $\text{\TeX}$ Maker: [www.xmlmath.net/texmaker](http://www.xmlmath.net/texmaker)
  - ▶  $\text{\TeX}$ nicCenter: <http://www.toolscenter.org>
  - ▶ Winshell: <http://www.winshell.de>
  - ▶ LEd: <http://www.latexeditor.org>
  - ▶  $\text{\TeX}$ Studio: [www.texstudio.org](http://www.texstudio.org)
  - ▶ VisualStudioCode: <https://code.visualstudio.com>

# Otthoni telepítés 2.

POSIX rendszerek alatt (Mac OS X, Linux, \*BSD stb.)

- ▶ a **T<sub>E</sub>Xlive** disztribúció ([www.tug.org/texlive](http://www.tug.org/texlive), csomagkezelőből telepíthető),
- ▶ Szövegszerkesztéshez:
  - ▶ Emacs: [www.gnu.org/software/emacs/emacs.html](http://www.gnu.org/software/emacs/emacs.html)
  - ▶ Kile: <http://kile.sourceforge.net/>
  - ▶ Nirvana Editor: [www.nedit.org](http://www.nedit.org)
  - ▶ **T<sub>E</sub>XStudio**: [www.texstudio.org/](http://www.texstudio.org/)

Multiplatform szerkesztők:

- ▶ T<sub>E</sub>XMaker: [www.xmlmath.net/texmaker/](http://www.xmlmath.net/texmaker/)
- ▶ VisualStudioCode: <https://code.visualstudio.com/>
- ▶ TeXstudio: [www.texstudio.org/](http://www.texstudio.org/)

Online szerkesztés:

- ▶ [www.overleaf.com](http://www.overleaf.com)

# Az első...

Az első  $\text{\LaTeX}$  kód, amit lefordítunk, a következő:

```
\documentclass{article}

\begin{document}
Hello world, bye bye Word!
\end{document}
```

Dok. szerk.  $\text{\LaTeX}$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléi alapján)

**Nulladik gyakorlat**

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

**Első gyakorlat**

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

Minden  $\LaTeX$  fájlban megtalálhatjuk az alábbi három fontos sort:

```
\documentclass{article}

\begin{document}
  ...Where do you want to go today?...
\end{document}
```

A forrásfájl felépítését tekintve két fő részből áll:

- ▶ Preambulum (a `\begin{document}` előtt)
- ▶ Dokumentumtörzs (a `\begin{document}` után)

# Magyar szöveg írása

Ha magyarul akarunk írni (ékezetes betűk), akkor egy külön csomagot be kell töltenünk:

```
\documentclass{article}
\usepackage[utf8]{inputenc}

\begin{document}
... Ide jön a szöveg ...
\end{document}
```

Figyelem! Bizonyos szövegszerkesztők nem UTF-8-ban mentik a fájlokat, hanem közép-európai kódolással. Ha így lenne, akkor `utf8` helyett `latin2`-t kell írni.

Furfangos karaktereket tartalmazó szöveg esetén szükség lehet még a `\usepackage[T1]{fontenc}` `\usepackage{lmodern}` csomagokra is, mellyel Seyðisfjörður, Đurđevac, Þórshöfn típusú szavak is leírhatók

(`Sey{\dh}isfj\"or{\dh}ur`, `{\DJ}ur{\dj}evac`, `\TH\'orsh\"ofn`).

A dokumentumosztályok típusai a következők (egyelőre maradjunk az első három egyikénél):

- ▶ article – cikk (általában ezt használjuk)
- ▶ report – beszámoló („könyvesebb”, mint az előző)
- ▶ book – könyv (ez már komoly dolog)
- ▶ letter – levél (hivatalos levélforma)
- ▶ slide – fólia (előadásokhoz)

# Dokumentumosztályok opciói

A dokumentumosztályokon belül léteznek opciók is, melyeket később fogunk részletezni. Az `a4paper` opció a standard A4-es kimenethez szükséges, használjuk mi is! A `babel` csomaggal magyaríthatjuk a  $\text{\LaTeX}$ et, erről később részletesen szó lesz, azért használjuk!

Példa:

```
\documentclass[a4paper]{report}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[latin2]{inputenc}
\usepackage[magyar]{babel}

\begin{document}
... Ide jön a szöveg ...
\end{document}
```



# Különleges karakterek

Speciális karakterek: `\`, `{`, `}`, `%`, `~`, `$`, `_`, `^`, `&`, `#`

Így írjuk őket: `\` = `\backslash`, `{` = `\{`, `}` = `\}`, `%` = `\%`,  
`~` = `\~{}`, `$` = `\$`, `_` = `\_`, `^` = `\^{}`, `&` = `\&`, `#` = `\#` A „fok”  
jelölésére speciális utasítás `\^{\circ}`: Odakint  $-50^{\circ}\text{C}$  van.

A `\`, `{`, `}`, `%` jelek feltétlenül fontosak a  $\text{\LaTeX}$ ben:

- ▶ Visszaper (vagy másnéven rep) jel: parancsok kezdete
- ▶ A kapcsos zárójelek blokkosítanak, például parancsok argumentumait is közéjük kell zárni
- ▶ A százalék kommentez, akár sorvége jelet is!

A százalék jellel (%) a sor hátralévő részét kikommentezhetjük. Még az újsor karaktert is figyelmen kívül hagyja a  $\LaTeX$ ! Sor elején lévő szóközők nem számítanak. A sor vége jel – ha csak egy van belőle – szóközőként funkcionál.

Így is írhatnám a szöveget, ha akarnám:

```
Így is írhatnám % de nem írom
a szöveg%
szöveg% szövjünk itt egy kicsit
v%
%
szöveget,
ha akarnám:
% lám itt nem volt szóköző, de volt újsor karakter
```



100 („kisel, nagyó, nagyó”)  $\neq$  100 (száz),

Rövid számok kiírása.

Szebb így: Kétszer kettő néha öt, mint  $2 \times 2$  néha 5!

Magyarban tizedesvessző, angolban tizedespont: 3,14, or 3.14. A helyiértékek jelölése magyarban térköz (pl. \, ), angolban vesszővel történik: 21 345,6 Ft, illetve 21,345.6 Euro, respectively.

Mértékegység, pénznemek: nem törhető szóközzel: 1~cm

1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm,  
1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm

Pont, vessző közvetlenül az őt megelőző szóhoz zárva, utána szóköz!!! Egyébként elég bután néz ki, ugye? Na.

Pontok halmozása: \dots, azaz ..., ami nem azonos a három ponttal (...), sem a „három pont és köztük szóközök”-kel (. . .)!!!

Zárójel szorosan zárja a környezetét (nincs szóköz), de kívülről szóközzel! Fordítva( ordítva )elég ocsmány.

Idézőjel: magyarban „ilyesmi” ( , , ilyesmi ’ ’ , két vesszővel nyitjuk és két aposztróffal zárjuk), britt angolban looks like ‘this’ ( ‘ this ’ , fordított és sima aposztrófok), in the US it “differs” ( ‘ ‘ differs ’ ’ , az előbbi duplán) a bit. A hüvelyk ( " , bár más tördelőkben, ahol nincs jobb ezt használtuk, itt nem jó!) nem idézőjel!!! Régebbi disztibuciókban az idézőjelek helyes megjelenítéséhez szükséges a t1enc csomag.

# Kötőjel, nagy kötőjel, gondolatjel, minuszjel, hosszú gondolatjel

Kötőjel (-): egy mínusz szimbólum leütésével előidézhető. Van-e kedved kipróbálni 5-6-szor?

Nagy kötőjel = gondolatjel (—): két mínusz szimbólum – mindkét oldalán szóközzel, vagy tartományt jelölve szóköz nélkül: 100–150 oldalas HF-eket fogadok el.

Hosszú gondolatjel (—, magyarban nem használjuk): angolban kedvelt, magyarban nem nyert. Három mínusz szimbólum.

Mínuszjel (-): \$-\$, odakint  $-5^{\circ}\text{C}$  van, odakint  $-5^{\circ}\text{C}$  van.

# Gondolatjel

Párbeszédet gondolatjellel kezdünk --\, szöveg alakban (nincs szóköz a --\, és a szöveg között!!!).

Hirtelen feltépte az ajtót, majd így kiáltott:

```
--\, Gyere elő a vackodból, te vén szipirtyó, --  
sziszegte dühösen -- ha nem jössz elő, megbúbollak%  
\dots
```

Még 10-12 alkalommal kiáltott a sötétbe (mert nem látta, hogy hol lehet az öregasszony a 4--8 méteres távolságban). Végül, „visszavonulót” fújt, mert nagyon fázott a  $-5^{\circ}\text{C}$ -os hidegben.

Hirtelen feltépte az ajtót, majd így kiáltott:

– Gyere elő a vackodból, te vén szipirtyó, – sziszegte dühösen – ha nem jössz elő, megbúbollak...

Még 10-12 alkalommal kiáltott a sötétbe (mert nem látta, hogy hol lehet az öregasszony a 4–8 méteres távolságban). Végül „visszavonulót” fújt, mert nagyon fázott a  $-5^{\circ}\text{C}$ -os hidegben.

# Elválasztási galibák

Bár a babel csomag az esetek többségében jól választ el, de van amit nem tudhat, pl. karóra (karó-ra, vagy kar-óra).

Lokális megkerülő megoldás. Puha elválasztójel alkalmazása, amely \- alakban adott. Pl. maka\-róni, makaróni, makaróni, makaróni, makaróni, makaróni, makaróni, makaróni

Globális megkerülő megoldás. A dokumentum tetszőleges részén kiadjuk a

```
\hyphenation{el-vá-lasz-ta-ni-mer-kell-neki-tyuhajja}
```

parancsot, és jól fog elválasztanimerkellnekiyuhajja elválasztanimerkellnekiyuhajja

Régebbi környezetekben, ha nincs betöltve a t1enc csomag, a \hyphenati on nem működik ékezetes karakteres szavakkal!



# Többsnyelvűség

Többsnyelvű doksik esetén a `\usepackage[english,magyar]{babel}` paranccsal indulunk, és a `\selectlanguage{nyelv}` paranccsal lehet kiválasztani az aktuális nyelvet.

Rövidebb, bekezdésen belüli idegen nyelvű sorokra a `\foreignlanguage{nyelv}{szöveg}` parancsot érdemes használni.

Létezik még a `foreignlangauge`-nek környezet változata is, azaz a

```
\begin{foreignlanguage}{nyelv}
szöveg
\end{foreignlanguage}
```

utasítások közé is lehet idegen nyelvű szöveget írni.

# Ligatúrák

A  $\LaTeX$  automatikusan kicseréli az ff, fl, fi és ffi betűkapcsolatok esetén a duplabetűket ligatúrákra. Ha nem akarjuk (miért nem?), akkor pl. `ef{}`féle elkerülő manőverekre van szükség.

Speciális karakterek: Ha magyaron kívül más nyelveken is akarunk szavakat írni, szükség lehet ezen utasításokra: `\'e = é`, `\'e = è`, `\~n = ñ`, `\"o = ö`, `\H{o} = ő`, `\~o = õ`, `\^u = û`, `\o = ø`, `\ss = ß` stb.

Ha i-re, vagy j-re teszünk ékezetet, akkor előbb le kell vennünk a pontot az `\i` és `\j` utasításokkal, amivel i és j betűket kapunk. Így már lehet jőpofa ékezetet tenni rájuk. De az í-t is így írjuk: `\'i`.

A  $\LaTeX$  tartalmaz néhány beépített utasítást, mellyel a  $\LaTeX$  logókat jeleníthetjük meg: `\TeX`, `\LaTeX` és `\LaTeXe`, azaz  $\TeX$ ,  $\LaTeX$  és  $\LaTeX_{\mathcal{E}}$ .

Alapértelmezésben a  $\LaTeX$  a mondatok közé nagyobb térközt tesz, mint a szavak közé. Ettől olvashatóbbá válik a nyomtatott szöveg. Ha nem akarjuk, a `\frenchspacing` paranccsal, vagy a `frenchspacing` csomag betöltésével kikapcsolhatjuk!

Honnan tudja a  $\LaTeX$ , hogy hol van a mondat vége? Ha kis betűt, vagy számot pont követ, akkor az mondatvég (nagy betű utáni pont általában rövidítés). Vannak azonban kivételek:

A 2. ábrán látható... Nem mondatvég, ezért A 2.~ábrán látható, vagy A 2.\ ábrán látható alakban kell írni.

Kedvencem a BME. Mondatvég, ezért Kedvencem a BME\@.  
alakban kell írni.

A szövegben kiemelést az `\emph{szöveg}` paranccsal lehet elérni. Hatására (alapbeállításként) *dőlt betűvel* jelenik meg a szöveg.

*Ha azonban a környezetében dőlt betűkkel szedtük a szöveget, akkor a kiemelés felállítja a kiemelt szavakat.* Azaz az `\emph{}` utasítás egymásba is ágyazható. `\emph{Akár hiszed \emph{akár} nem}`

Lábjegyzetbe<sup>1</sup> a `\footnote{szöveg}` parancs használatával írunk. Hatására a szöveg a lap alján, apróbb betűvel jelenik meg.

A `\footnote` parancsot közvetlenül – szóköz nélkül – ahhoz a szóhoz, vagy mondathoz kötjük, amelyről írni akarunk. Például<sup>2</sup> így. Például `\footnote{elegem van a példákból}` így.

---

<sup>1</sup> azért lábjegyzet, mert a lap alján jelenik meg apróbb betűvel

<sup>2</sup> elegem van a példákból

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: Angol és magyar nyelvű részeket tartalmazó dokumentum tördelése, amelyben mindkét nyelven szerepelnek számok mértékegységgel, idézetek és elválasztások.

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 1 000–10 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje az óra vége.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

A  $\LaTeX$  használatának alapjai

## A szöveg begépelése

### A dokumentum struktúrálása

Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

Táblázatok, ábrák

Táblázatok

Képek, ábrák

Matematikai képletek I

Matematikai képletek II

Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

Térközök, számlálók

Vízszintes és függőleges térközök

Számlálók

Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

Dobozok

Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

**Második gyakorlat**

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

A  $\LaTeX$  használatának alapjai

## A szöveg begépelése

### A dokumentum struktúrálása

Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

Táblázatok, ábrák

Táblázatok

Képek, ábrák

Matematikai képletek I

Matematikai képletek II

Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

Dobozok

Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

**Második gyakorlat**

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# A dokumentum struktúrálása (fejezetek, alfejezetek)

Új bekezdéshez egy üres sor legyen a forrásban.

```
\part{Rész cím} -> csak book és report osztályokban  
\chapter{Alrész cím} -> csak bookban és reportban  
\section{Fejezet címe}  
\subsection{Alfejezet címe}  
\subsubsection{Al-alfejezet címe}  
\paragraph{Még kisebb egység címe}  
\subparagraph{A legkisebb egység címe}
```

Ugyanezek csillaggal (`\section*{Hello}` stb.): számozatlan fejezetek

A `\tableofcontents` utasítás automatikusan generált tartalomjegyzéket illeszt be. Csak a számozott fejezeteket tartalmazza.



# Dokumentum címe, szerző stb.

A `\title{...}`, `\author{...}`, `\date{...}` utasításokkal megadhatjuk a dokumentum címét, a szerzőt és a dátumot. A `\date{}` utasítás beillesztésével érhetjük el, hogy ne szerepeljen dátum a címben. A `\date{\today}` utasítás olyan, mintha nem is lenne: ha elmarad a `\date`, akkor a rendszeridő alapján a mai dátumot állítja be a  $\LaTeX$ .

Több szerző esetén az `\and` paranccsal sorolhatjuk fel őket. A cím beillesztése a `\maketitle` paranccsal lehetséges.

A dokumentum bármely részén ki lehet írni az aktuális dátumot, a `\today` paranccsal. Ma pl. 2024. február 14. van.

# Egy példa

```
\documentclass[a4paper]{book}
\usepackage[latin2]{inputenc}
\usepackage{t1enc}
\usepackage[magyar]{babel}

\begin{document}
\title{Az aranyhalak nemi élete \\\
és egyéb cigerettatípusok}
\author{Dr.~Kiss Géza \\\ egyetemi tanár \and
Kovács Alajos \\\ nyugdíjas alezredes}
\maketitle
\chapter{Bevezetés}
\section{Előzmények}
Itt jön a szöveg maga\ldots
\end{document}
```

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: Minnél több módosított latin karakter (pl. magyar, lengyel, skandináv, vietnámi) megjelenítése kizárólag ascii tex forrással (internet segítségével). A dokumentumnak legyen címe (maketitle).

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 1 000–10 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje az óra vége.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

## Alapvető környezetek

**Programkód beillesztése  $\LaTeX$  forrásba**

**Felsorolások. Leíró, számozott és számozatlan listák**

**Idézetek, versek környezetei**

**Irodalomjegyzék és arra hivatkozás**

## Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

## Táblázatok, ábrák

Táblázatok

Képek, ábrák

## Matematikai képletek I

## Matematikai képletek II

## Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

## Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

## Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

## Dobozok

## Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

## További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

**Harmadik gyakorlat**

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Környezetekről általában

$\LaTeX$  alatt a valami környezetet `\begin{valami}` utasítással nyitjuk meg, és `\end{valami}` utasítással zárjuk.

Eddig is használtuk a `document` környezetet.

Környezetek egymásba is ágyazhatóak, de csak teljes átfedéssel.

Azaz a

```
\begin{aaa} ... \begin{bbb} ... \end{bbb} ... \end{aaa}
```

előfordulhat, de

```
\begin{aaa} ... \begin{bbb} ... \end{aaa} ... \end{bbb}
```

nem létezhet.

Az utasítások `\command{param}` alakúak. A  $\LaTeX$  mindenképpen vár paramétert, ezért kell trükköznünk a paraméter nélküli utasításoknál (pl. `\LaTeX` és `\dots`). A szögletes zárójelekbe írt paraméterek opcionálisak (lásd pl. a `\documentclass[a4paper]{book}` parancsot.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

## Alapvető környezetek

### Programkód beillesztése $\LaTeX$ forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

## Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

## Táblázatok, ábrák

Táblázatok

Képek, ábrák

## Matematikai képletek I

## Matematikai képletek II

## Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

## Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

## Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

## Dobozok

## Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

## További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

**Harmadik gyakorlat**

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

Ha programkódot kell megjeleníteni egy doksiban és abban sok, a  $\LaTeX$  által vezérlőkarakternek használt karakter található, akkor ne menjen el a kedvünk, létezik megoldás!

A `\verb+szöveg\@%+` utasítással a `szöveg\@%` verbatim módban jelenik meg, azaz minden karakter megjelenik. Jelen esetben a `+` jellel jelöltük a verbatim széleit, de tetszőleges más karakter is használható (kivéve a betűket a szóközt és a csillagot)

Ha hosszabb – több soros – verbatimot szeretnénk, akkor célszerű a `\begin{verbatim}` `szöveg` `\end{verbatim}` környezetet használni.

A fenti utasítás és környezet csillagos változata megjeleníti a szóközöket is, pl. `\verb*+Helló világ!+` hatására `Helló␣világ!` kimenetet láthatjuk.

# Istlisting – az intelligensebb megoldás

A `listings` csomag betöltése után (a preambulumba a `\usepackage{listings}` utasítást kell írni) szebben is tördelhetjük programkódjainkat.

Például nyelv specifikus, tartalom kiemelés, font változatokkal (és színekkel):

[https://www.sharelatex.com/learn/Code\\_listing](https://www.sharelatex.com/learn/Code_listing)



# Pseudo kód – algorithm csomag családdal

Ha nem egy specifikus program nyelv megjelenítésére, hanem egy általános algoritmus leírására van szükség akkor alkalmazható az algorithm csomag.

<https://en.wikibooks.org/wiki/LaTeX/Algorithms>

Például az algorithmicx és az algpseudocode csomagokkal:

```
if  $i \geq maxval$  then  
     $i \leftarrow 0$   
else  
    if  $i + k \leq maxval$  then  
         $i \leftarrow i + k$   
    end if  
end if
```

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

## Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

### Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

## Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

## Táblázatok, ábrák

Táblázatok

Képek, ábrák

## Matematikai képletek I

## Matematikai képletek II

## Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

## Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

## Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

## Dobozok

## Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

## További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

**Harmadik gyakorlat**

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

## Felsorolások, listák 2.

Három típusú felsorolás létezik  $\LaTeX$  alatt: a számozatlan lista (`itemize`), a számozott lista (`enumerate`) és a leíró lista (`description`). Mindhárom környezetként létezik.

```
\begin{itemize}
\item Első elem
\item Második elem
\item Harmadik elem
\end{itemize}
```

Ugyanez `itemize` helyett `enumerate`-tel számozottá válik.

## Felsorolások, listák 2.

Ha a felsorolás egy pontját ki akarjuk emelni, akkor használhatjuk az `\item` parancs opcionális paraméterét (figyelem! Erre nagyon ritkán van szükség!). Például:

```
\begin{itemize}
\item Első elem
\item[$\circ$] Második, kiemelt elem
\end{itemize}
```

A `description` környezetben a leírt kifejezést az opcionális paraméterben adjuk meg, pl.

```
\begin{description}
\item[a busa husa] rendkívül finom,
\item[a süllő sült] nagyon ízletes.
\end{description}
```

# Felsorolások, listák egymásba is ágyazhatóak

```
\begin{enumerate}
\item Első külső elem
\item Második külső elem
\begin{enumerate}
\item Első belső elem
\item Második belső elem
\begin{description}
\item[a butaságok] listában sem válnak tudományossá,
\item[a jó gondolatok] szebben mutatnak rendszerezve
\end{description}
\item Harmadik belső elem
\end{enumerate}
\item Harmadik külső elem
\end{enumerate}
```

## 1. Első külső elem

## 2. Második külső elem

### 2.1 Első belső elem

### 2.2 Második belső elem

**a butaságok** listában sem válnak tudományossá,  
**a jó gondolatok** szebben mutatnak rendszerezve

### 2.3 Harmadik belső elem

## 3. Harmadik külső elem

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

## Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

### Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

## Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

## Táblázatok, ábrák

Táblázatok

Képek, ábrák

## Matematikai képletek I

## Matematikai képletek II

## Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

## Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

## Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

## Dobozok

## Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

## További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

**Harmadik gyakorlat**

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Idézetek

A  $\LaTeX$  három környezetet támogat az idézetek beszúrásához. Ezek a `quote`, a `quotation` és a `verse` környezetek.

A `quote` környezetben nem kezdődnek behúzással a bekezdések. A `quotation` környezetben a bekezdések behúzással kezdődnek, míg a `verse` környezetben minden sor eleje azonos helyen kezdődik, a sorok végét `\\` karakterekkel jelöljük.

```
\begin{verse}
Tűz, te gyönyörű \\
lobogó, csillag erejű \\
Te fűtsd a mozdonyt halálra \\
Hajszold, hogy fekete magánya \\
Ne legyen néki teher.
\end{verse}
```

*Tűz, te gyönyörű  
lobogó, csillag erejű  
Te fűtsd a mozdonyt halálra  
Hajszold, hogy fekete magánya  
Ne legyen néki teher.*

# Példa az idéző környezetek használatára

quote és quotation hosszú sorokkal rendelkező szövegrészleten:

```
\begin{quote}
```

```
Ha elég hosszúak a sorok, akkor megláthatjuk, miféle  
különbséget rejt e két környezet, persze figyelniük  
kell arra is, hogy egy új bekezdést is kezdjünk.
```

```
Ha elég hosszúak a sorok, akkor megláthatjuk, miféle  
különbséget rejt e két környezet, persze figyelniük  
kell arra is, hogy egy új bekezdést is kezdjünk.
```

```
\end{quote}
```

*Ha elég hosszúak a sorok, akkor megláthatjuk, miféle különbséget rejt e két környezet, persze figyelniük kell arra is, hogy egy új bekezdést is kezdjünk.*

*Ha elég hosszúak a sorok, akkor megláthatjuk, miféle különbséget rejt e két környezet, persze figyelniük kell arra is, hogy egy új bekezdést is kezdjünk.*



# Példa az idéző környezetek használatára

quote és quotation hosszú sorokkal rendelkező szövegrészleten:

```
\begin{quotation}
```

Ha elég hosszúak a sorok, akkor megláthatjuk, miféle különbséget rejt e két környezet, persze figyelniük kell arra is, hogy egy új bekezdést is kezdjünk.

Ha elég hosszúak a sorok, akkor megláthatjuk, miféle különbséget rejt e két környezet, persze figyelniük kell arra is, hogy egy új bekezdést is kezdjünk.

```
\end{quotation}
```

*Ha elég hosszúak a sorok, akkor megláthatjuk, miféle különbséget rejt e két környezet, persze figyelniük kell arra is, hogy egy új bekezdést is kezdjünk.*

*Ha elég hosszúak a sorok, akkor megláthatjuk, miféle különbséget rejt e két környezet, persze figyelniük kell arra is, hogy egy új bekezdést is kezdjünk.*

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

## Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

## Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

## Matematikai képletek I

## Matematikai képletek II

## Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

## Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

## Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

## Dobozok

## Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

## További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

**Harmadik gyakorlat**

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Az irodalomjegyzék beillesztése

A thebibliography környezettel lehetséges. A környezetnek van egy argumentuma, amely segíti a  $\LaTeX$ -et az irodalomjegyzék tördelésében. Egyelőre elég annyit tudni, hogy ha tíznél kevesebb művet sorolunk fel, akkor egy 9-est, ha száznál kevesebbet, de tíznél többet, akkor 99-et stb. kell ide írni.

A hivatkozott irodalmakat a `\bibitem{cimke}` paranccsal soroljuk fel (a `\cite{cimke}` utasítással tudunk majd rá hivatkozni). Az irodalomjegyzék automatikusan generálódik. Alapbeállításként a sorszámok növekvő arab számok szögletes zárójelek között.

A `\bibitem[Jópofa]{cimke}` hatására azonban a tartalomjegyzék vonatkozó elemének Jópofa lesz a „száma”, és ha hivatkozunk rá a szövegben, akkor is a [Jópofa] jelenik meg.

# Az irodalomjegyzék szabályos (IEEE) alakja

A szabályosan szedett irodalmi hivatkozások alakja a következő:

- ▶ Könyv: szerző, *cím dőlt betűkkel*, kiadó neve, kiadó városa, kiadás éve.
- ▶ Újságcikk: szerző, „cím idézőjelek között”, *újság neve dőlt betűkkel*, évfolyam, szám, oldalak, kiadás éve
- ▶ Diplomadolgozat/disszertáció: szerző, *cím dőlt betűkkel*, oktatási intézmény, város, végzés éve.
- ▶ elektronikus forrás: teljes URL (átviteli protokoll megjelölésével), megtekintés időpontja percre pontosan

Példa:

```

...itt pedig hivatkozok a könyvre~\cite{konyv}.
...
\begin{thebibliography}{9}
\bibitem{konyv}
Virágvölgyi Péter,
{\it A tipográfia mestersége},
Osiris, Budapest, 2001.
\bibitem{cikk}
Claude E.\ Shannon,
,,A Mathematical Theory of Communication'',
{\it Bell System Technical Journal}, vol.~27, no.~10,
pp.~379--423 és pp.~623--656, 1948.
\bibitem{url}
{\tt http://www.karinty.hu/~{}NIKITA/szedes/Bevezet.htm},
2002.~január~9., 19:14
\end{thebibliography}

```

A `\cite` utasításnak létezik opcionális paramétere is, amellyel például az oldalszámot (vagy egyéb kiegészítő információt) adhatjuk meg. A kimeneten az irodalom címkéje mellett vesszővel jelenik meg a szögletes zárójelen belül [1, 160–165. oldal].

Példa:

```
...zárójelen belül~\cite[160--165.\ oldal]{konyv}.
...
\begin{thebibliography}{9}
\bibitem{konyv}
  Virágvölgyi Péter,
  {\it A tipográfia mestersége},
  Osiris, Budapest, 2001.
...
\end{thebibliography}
```

## Bibtex adatbázis létrehozása (valami.bib):

```
@article{Kurtz70,  
author = "Thomas G. Kurtz",  
title = "Solutions of Ordinary Differential Equations as Limits of Pure  
journal = "Journal of Applied Probability",  
volume = "7",  
number = "1",  
pages = "49 - 58",  
year = "1970"  
}
```

```
@BOOK{pilar,  
Author = {Frank L. Pilar},  
title = "Elementary Quantum Chemistry",  
publisher = "McGraw-Hill",  
edition = "Second",  
year = 1990,  
}
```

Hivatkozás irodalomra ugyan úgy mint eddig:

```
... \cite{pilar}  
... \cite[Sec. 2]{Kurtz70}
```

Irodalomjegyzék beszúrása:

```
\bibliography{valami}  
\bibliographystyle{plain}
```

Latex fordítás:

```
pdflatex forras[.tex]  
bibtex forras[.aux] (-> .bbl)  
pdflatex forras[.tex]  
pdflatex forras[.tex]
```



## Bibliográfia stílusok:

style	author	reference	sorting
plain	full name	[1]	by atuhor
unsrt	full name	[1]	as referenced
abbrv	short name	[1]	by atuhor
alpha	full name	short name,year	by atuhor
apacite	full name (year)	name, year	by atuhor
acm, apalike, ieetr, siam, ...			

[https://www.sharelatex.com/learn/Bibtex\\_bibliography\\_styles](https://www.sharelatex.com/learn/Bibtex_bibliography_styles)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

## Bibtex adatbázis a 93. oldali bbl-hez:

```
@book{konyv,  
author= {Virágvölgyi Péter},  
title={A tipográfia mestersége},  
publisher={Osiris}, address={Budapest},  
year={2001},  
}  
  
@article{cikk  
author= {Claude E. Shannon},  
title={A Mathematical Theory of Communication},  
journal={Bell System Technical Journal},  
volume={27}, number={10},  
pages={379--423}, year={1948},  
}  
  
@misc{url,  
author = {}, title = {},  
howpublished=\url{http://www.karinthy.hu/~{}NIKITA/szedes/Bevezet.htm},  
note = {accessed 2002. Janu\’ar 9, 19:14},  
}
```

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: A

<http://webspn.hit.bme.hu/telek/cikkek/horv05c.pdf> cikk  $\LaTeX$ be

írása a bevezetés első felsorolásának végéig a hivatkozásokkal

(bibtex) és címmel együtt, kivonat nélkül article stílusban.

Opcionális feladat ugyan ez IEEE transaction (kéthasábos) stílusban.

A házi feladatban a bruttó (utasításokkal együtt számolt)

karakterszám 10 000–15 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje az óra vége.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

**Negyedik gyakorlat**

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

## Hivatkozások, utasítások, környezetek

**Hivatkozás nem irodalomjegyzék elemre**

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

## Táblázatok, ábrák

Táblázatok

Képek, ábrák

## Matematikai képletek I

## Matematikai képletek II

## Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

## Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

## Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

## Dobozok

## Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

## További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

**Negyedik gyakorlat**

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Automatikus hivatkozások 1.

A  $\LaTeX$  automatikus metódust kínál a hivatkozások terén. Minden számozott objektumra lehet hivatkozni. Módja egyszerű: a hivatkozni kívánt objektumot megcímkézzük a `\label{cimke}` utasítással, majd a hivatkozás helyén a `\ref{cimke}` utasítással hívjuk elő a számot.

Példa:

```
\section{Bélabácsi}
```

```
\label{sec:bela}
```

```
Itt kezdem a fejezetet\dots
```

```
...
```

```
Később (vagy akár hamarabb is) elkezdek hivatkozni  
a~\ref{sec:bela}.~fejezetre.
```

## Automatikus hivatkozások 2.

Mindenre hivatkozhatunk, ami számozott (lábjegyzetre, listaelemre, fejezetre, képletre, táblázatra stb.). Oldalra a `\pageref{cimke}` utasítással hivatkozhatunk. Példa:  
Bővebben a `\pageref{sec:bela}`.~oldalón...

Nem kötelező, de gyakran követett konvenció, hogy a címkében megjelöljük az objektum típusát. Például fejezeteket, alfejezeteket `sec:xxx`, táblázatokat `tab:xxx`, ábrákat `fig:xxx`, képleteket `eq:xxx` címkével stb. látjuk el.

Irodalomjegyzék belső elemre a `\cite{cimke}` utasítással hivatkozunk, a `\ref{}` nem használható! Oka: külön kezeljük az irodalmi hivatkozások címkeit pl., hogy a Bi<sub>T</sub>E<sub>X</sub> ne keverje össze őket.

# Határozott névelő a hivatkozás előtt

Az `\az{valami}` és `\Az{valami}` utasításokkal a `valami` elé határozott névelőt tehetünk, a különbség: az első kisbetűvel írja a névelőt, a második nagybetűvel kezdi (mondat elején jó). Az `\aref{}` és `\Aref{}` utasításokkal névelővel hivatkozhatunk objektumokra (tehát gyakorlatilag `\aref{cimke} = \az{\ref{cimke}}`).

Az `\acite{}`, `\Acite{}`, `\apageref{}` és `\Apageref{}` utasítások hasonlóan az előzőek kiegészítései. A korábbi példák tehát helyesen:

Később (vagy akár hamarabb is) elkezdek hivatkozni `\aref{sec:bela}.~fejezetre`.  
Bővebben a Bélabácsi fejezetben, `\apageref{sec:bela}.~oldalon...`



# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

### Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

**Negyedik gyakorlat**

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Új utasítás létrehozása

A `\newcommand` paranccsal történik. Az első argumentumba kell beírni az utasítás azonosítóját, a második argumentumban pedig a végrehajtandó parancsokat kell leírni.

Például a `\newcommand{\ft}{\emph{FONTOS!!!}}` hatására minden `\ft` utasítás a *FONTOS!!!*-t eredményezi. Ha már létező parancsot akarunk felüldefiniálni, akkor a `\renewcommand` parancsot kell alkalmazzuk, például így:

`\renewcommand{\ft}{\emph{nem is olyan fontos\dots}}`, hatására az `\ft` utasítás már a *nem is olyan fontos...-t* eredményezi.

Ha egy esetleg létező utasítást akarunk definiálni, de csak akkor, ha nem létezik, akkor a `\providecommand` utasításra van szükségünk.

# Új utasítás létrehozása argumentumokkal

A definiált parancsok argumentumokat is fogadhatnak, ekkor a `\newcommand{\prncs}[argsz][default]{def}` alakot kell alkalmazni. Az utasítással a `\prncs` parancsot definiáljuk, melynek `argsz` számú argumentuma van. Az első argumentum opcionálissá tehető. Az első argumentum alapértelmezett beállítását a `default` írja le. A parancs definíciója a `def` argumentumban található. Az argumentumokat a `#1`, `#2` stb. alakban hívhatjuk elő.

Ha a parancs argumentuma nem lehet több egy bekezdésnél akkor a csillagos verziókat használjuk (`\newcommand*`, `\renewcommand*` stb.).

Példa: automatikus hivatkozás fejezetre és használata:

```
\newcommand*{\secref}[1]{\aref{sec:#1}.~fejezet}  
\newcommand*{\Secref}[1]{\Aref{sec:#1}.~fejezet}  
\Secref{bela}ben olvashatunk arról, hogy...
```

# Új környezet létrehozása

Hasonlóképpen hozhatunk létre környezeteket a `\newenvironment{körny}[argsz][def]{nyitódef}{záródef}`, illetve a `\renewenvironment{körny}[argsz][def]{nyitódef}{záródef}` parancsokkal. Hatásukra a `körny` nevű környezetet `argsz` számú argumentummal definiáljuk. A csillagos verziókat érdemes használni, ha az argumentumok nem lehetnek hosszabbak egy bekezdésnél.

Példa:

```
\newenvironment*{bela}%  
{Nyitottam egy környezetet, azt írtam bele, hogy}%  
{Elég már ebből a környezetből, bezárom itt.}  
\begin{bela}  
Kacsacsőr  
\end{bela}
```

Nyitottam egy környezetet, azt írtam bele, hogy Kacsacsőr Elég már ebből a környezetből, bezárom itt.

# Verseskötet példa

Másik példa, mellyel verseskötetet lehet egyszerűen szedni:

```
\newenvironment*{vers}[2][Petőfi Sándor]%
{\begin{center} #1: {\sc #2} \end{center}}
\begin{verse}}{\end{verse}}
\begin{vers}[Arany János]{Toldi}
Ég a napmelegtől a kopár szík sarja \\
...
\end{vers}
\begin{vers}{Anyám tyúkja}
Ej, mi a kő! Tyúkanyó, kend \\
...
\end{vers}
```

Arany János: TOLDI

*Ég a napmelegtől a kopár szík sarja*

...

Petőfi Sándor: ANYÁM TYÚKJA

*Ej, mi a kő! Tyúkanyó, kend*

...

# Ha a környezet argumentumát hátul szeretnénk használni...

Ha az egyik parancsargumentumot a záró definícióban akarjuk alkalmazni, akkor trükközni kell. A `\newsavebox{\valami}` utasítással létre kell hozni egy string tárolására alkalmas `\valami` változót, melynek a `\sbox{\valami}{érték}` utasítással adhatunk értéket. A későbbiekben a `\usebox{\valami}` utasítással írathatjuk ki a `\valami` értékét.

Az előbbi környezet a költő nevének aláírásával:

```
\newsavebox{\kolto}
\newenvironment*{vers}[2][Petőfi Sándor]%
{\begin{center}{\sc #2}\end{center}
\sbox{\kolto}{#1}\begin{verse}}%
{\end{verse}\begin{flushright}\usebox{\kolto}
\end{flushright}}
```

A környezetet természetesen ugyanúgy használjuk, mint az előbb.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

### Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

**Negyedik gyakorlat**

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Tárgymutató készítése

Sokszor szükséges tárgymutató egy dokumentumhoz, megkönnyíti az olvasó munkáját, ha szöveges kulcsra akar rákeresni.

Használatához a preambulumban a `\usepackage{makeidx}` és `\makeindex` parancsokat kell kiadni. A tárgymutató tartalmát a dokumentum testében lévő `\index{kulcs}` kulcsok fogják alkotni. A `\printindex` utasítással a dokumentum tetszőleges helyén kinyomtathatjuk a tárgymutatót. Az `\index{kulcs!alkulcs}` utasítással albejegyzést is lehet csinálni.

Ha a tartalomjegyzékbe is be szeretnénk írni akkor

```
\clearpage
```

```
\addcontentsline{toc}{chapter}{Tárgymutató}
```

```
\printindex
```



# Tárgymutató fordítása

A `\makeindex` utasítás hatására az összes kulcs kiíródik a `valami.idx` fájlba, ha a `valami.tex` fájlba dolgozunk. A `\printindex` parancs a `valami.ind` fájl tartalmát írhatja ki. A `valami.ind` a `valami.idx` fájl sorba rendezett változata, amit pl. a `makeindex valami` utasítással érhetünk el. Ez utóbbit bizonyos környezetekben a terminálban kell futtatni.

Fordítás:

```
pdflatex forras[.tex] (->.idx)
makeindex forras[.inx] (->.ind)
pdflatex forras[.tex]
pdflatex forras[.tex]
```

# (pdf)latex fordításkor keletkező index és forrás fileok

parancs	szükséges csomag	generált file	
- (mindig)	-	*.aux	index
<code>\tableofcontents</code>	-	*.toc	forras
<code>\listoffigures</code>	-	*.lof	forras
<code>\listoftables</code>	-	*.lot	forras
<code>\makeindex</code>	makeindex	*.idx	index
<code>\makeglossaries</code>	glossaries	*.glo	index

# forrás fileok generálása index fileok alapján

miktex parancs	index file	generált file
bibtex	*.aux	*.bbl
makeindex	*.idx	*.ind
makeglossaries	*.glo	*.gls

ha `makeglossaries` nem elérhető, akkor

```
makeindex *.glo -o *.gls -s glossaries.ist
```

# forrás fileok includolása (pdf)latex fordításkor

parancs	includált file
<code>\bibliography</code>	<code>*.bbl</code>
<code>\tableofcontents</code>	<code>*.toc</code>
<code>\listoffigures</code>	<code>*.lof</code>
<code>\listoftables</code>	<code>*.lot</code>
<code>\printindex</code>	<code>*.ind</code>
<code>\printglossaries</code>	<code>*.gls</code>

Csak ha az adott file létezik!

Ha a file nem létezik nincs hibaüzenet, mivel első fordításnál még üzemszerűen nem létezik az adott file!!!

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: index lista készítése. Pontos specifikáció az órán

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 10 000–15 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje az óra vége.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

## Táblázatok, ábrák

### Táblázatok

## Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközök, számlálók

Vízszintes és függőleges térközök

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

**Ötödik gyakorlat**

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

## Táblázatok, ábrák

### Táblázatok

Képek, ábrák

## Matematikai képletek I

## Matematikai képletek II

## Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

## Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

## Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

## Dobozok

## Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

## További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

**Ötödik gyakorlat**

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

A  $\LaTeX$  alatt a `tabular` környezettel illeszthetünk be táblázatot. A környezet argumentumában adjuk meg, hogy milyen oszlopai lesznek a táblázatnak, illetve azt, hogy mely oszlopok közé szeretnénk függőleges vonalat húzni (`|`). Az oszlopok négy típusba sorolhatók: `left` (balra igazított), `right` (jobbra igazított), `center` (középre igazított) és `p{szélesség}`: adott szélességű.

Példa két oszlopú, három csíkos táblázatra:

```
\begin{tabular}{|l|r|}
balra igazított oszlop & jobbra igazított oszlop \\
akkor látszik & ha több elemű \\
\end{tabular}
```

balra igazított oszlop	jobbra igazított oszlop
akkor látszik	ha több elemű



## Táblázatok 2

A táblázatban a sorokat `\` jellel választjuk el egymástól, a soron belüli elemeket pedig az `&` jellel. Amelyik sor elé vízszintes vonalat szeretnénk húzni, ott a `\hline` parancsot kell kiadjunk. Így ha az első sorban és az utolsóban szerepel a `\hline` utasítás, akkor keretezett lesz a táblázat.

Példa kétszer kettes keretezett táblázatra:

```
\begin{tabular}{|r|l|}  
\hline  
Egy & megérett a meggy \\  
Kettő & csipkebokor vessző \\  
\hline  
\end{tabular}
```

Egy	megérett a meggy
Kettő	csipkebokor vessző

## Táblázatok 3

Dupla vonalakat a `|` és a `\hline` duplázásával lehet elérni. Ha több, azonos rendű oszlopunk van, akkor a deklaráció lerövidíthető a `*` operátorral: `a *{3}{r|l|c|}` ekvivalens az `r|l|c|r|l|c|r|l|c|` definícióval. Példa nyolc vonatos vasúti menetrendre:

```
\begin{tabular}{l||*8{c|}}
```

```
A vonat száma:                & 437 & ... \\
```

```
\hline\hline
```

```
Indul Bp.\ Keleti-pu.-ról      & 8:00 & ... \\
```

```
Érkezik Hatvanba              & 8:52 & ... \\
```

```
Indul Hatvanból               & 8:55 & ... \\
```

```
Érkezik Miskolc Tiszai-pu.-ra & 10:22 & ... \\
```

```
\hline
```

```
\end{tabular}
```

A vonat száma:	437	...
Indul Bp. Keleti-pu.-ról	8:00	...
Érkezik Hatvanba	8:52	...
Indul Hatvanból	8:55	...
Érkezik Miskolc Tiszai-pu.-ra	10:22	...

# Táblázatok 4

Az @ paraméterrel a táblázat oszlopai közötti elválasztó átírható; a @{} nullára csökkenti az oszlopok közötti távolságot, a @{:} kettőspontot tesz két oszlop elemei közé. A \multicolumn{n}{oszldef}{szöveg} paranccsal n cellát összevonva oszldef módon írjuk a szöveget.

Példa: menetrend szebben (a kettőspontok egymás alá kerülnek):

```
\begin{tabular}{l||*8{r@{:}l|}}
A vonat száma: & \multicolumn{2}{c|}{437} &
... \\ \hline\hline
Indul Bp.\ Keleti-pu.-ról & 8&00 & ... \\
Érkezik Hatvanba & 8&52 & ... \\
Indul Hatvanból & 8&55 & ... \\
Érkezik Miskolc Tiszai-pu.-ra & 10&22 & ... \\
\hline
\end{tabular}
```

## Táblázatok 5

A `\multicolumn{1}{oszlof}{szöveg}`-gel egyetlen cellában írhatjuk felül az oszlopokra vonatkozó definíciót. Az `@{}` nagyon hasznos, mert ismétlődő mintákat könnyedén beilleszthetünk az oszlopok elé és után. A tradicionális nyomdászatban számok írásakor az azonos helyiértékeknek egymás alá kell kerülniük. Ez csak akkor probléma, ha tizedestörteket kell egymás alá írni. A megoldás: `r@{,}1`. Példa táblázatra számokkal:

```
\begin{tabular}{c|r@{,}1}
Kifejezés      & \multicolumn{2}{c}{Érték} \\
\hline
 $\pi$           & 3&1415927      & \\
 $\pi^{\pi}$        & 36&46216       & \\
 $\pi^{\pi^{\pi}}$  & 80662&666      & \\
\end{tabular}
```

Kifejezés	Érték
$\pi$	3,1415927
$\pi^{\pi}$	36,46216
$\pi^{\pi^{\pi}}$	80662,666



## Táblázatok 7

Ha csak adott oszlopokban szeretnénk a cellák fölé vonalat húzni, akkor a `\hline` helyett a `\cline{x-y}` utasítás kell, ahol  $x$  és  $y$  a vonal két vége (oszlopban). Az oszlopok számozása egytől kezdődik. Példa:

```
\begin{tabular}{l|r@{,}l@{\,}Ft\ \ }
\multicolumn{1}{c|}{Áru neve} &
\multicolumn{2}{c}{Érték} \\
\hline
Krumpli & 24&40 \\
Kenyér & 106&20 \\
Tej & 110&80 \\
\cline{2-3}
Összesen & 231&40
\end{tabular}
```

Áru neve	Érték
Krumpli	24,40 Ft
Kenyér	106,20 Ft
Tej	110,80 Ft
Összesen	231,40 Ft

A táblázatok elég bután néznek ki szövegben, ezért célszerű „úsztatni” őket. Erre szolgál a `table` környezet. A `table` környezettel „be kell burkolni” a `tabular` környezetet. A `table` környezetnek van egy opcionális paramétere, mellyel befolyásolhatjuk a táblázat helyét, lehetséges értékei here (ide vele), top (lap tetejére), bottom (lap aljára), page (külön lapra). Az opcionális paraméterben ezek tetszőleges sorrendben felsorolhatóak, mellyel a prioritásukat határozzuk meg. A  $\LaTeX$  kiválasztja melyik a legjobb és elhelyezi a táblázatot. Alapértelmezés: `[tbp]`. Ha azt szeretnénk, hogy mindenképpen adott helyre tegye, akkor egy felkiáltójellel lehet szigorítani az utasítást: `[!h]` jelentése: mindenképpen ide tedd.

A `table` környezet belsejében a `\caption{szöveg}` paranccsal adhatunk feliratot a táblázathoz, illetve itt is lehet a `\label{cimke}` parancsot használni. A `\caption` pozíciójától függően a táblázat felett, vagy alatt jelenik meg a felirat.

# Példa úsztatott táblázatra

```
\begin{table}[htb]
\caption{Nagy bölcsességek}
\label{tab:fontos}
\center
\begin{tabular}{c|r@{,}l}
Kifejezés      & \multicolumn{2}{c}{Érték} \\ \hline
 $\pi$           & 3&1415927      & \\
 $\pi^{\pi}$       & 36&46216      & \\
 $\pi^{\{\pi^{\pi}\}}$  & 80662&666      & \\
\end{tabular}
\end{table}
```



# Az előbbi egy új környezet definiálásával

```
\newenvironment*{tabl}[3]
% #1: caption, #2: cimke, #3: oszlopdef
{\begin{table}[htb]%
\caption{#1} \label{tab:#2} \center%
\begin{tabular}{#3}}
{\end{tabular}\end{table}}
...
\begin{tabl}{Nagy bölcsességek}{fontos}{c|r@{,}l}
Kifejezés      & \multicolumn{2}{c}{Érték} \\
\hline
 $\pi$  & 3&1415927 & \\
 $\pi^{\pi}$  & 36&46216 & \\
 $\pi^{\{\pi^{\pi}\}}$  & 80662&666 & \\
\end{tabl}
```

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

## Táblázatok, ábrák

Táblázatok

## Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközök, számlálók

Vízszintes és függőleges térközök

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

**Ötödik gyakorlat**

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Ábrák készítése

A  $\LaTeX$ -nek van beépített parancskészlete ábrák készítéséhez, de mára elavult. Helyette latex környezetben a `tikz` csomag alkalmazása terjed mostanában.

Nem latex környezetben grafikus felülettel rendelkező programok használhatóak ábra készítésre, majd EPS (Encapsulated PostScript) vagy PDF formátumba mentésre.

Ábrák egy további potenciális forrása az internet.

Más formátumú ábrák PDF konvertálása pl. nyomtató driverekkel lehetséges, vagy `ps2pdf -dEPSCrop *.eps *.pdf`.

**PDF $\LaTeX$  fordítás esetén alapértelmezésben csak a PDF ábrák működnek.**

Ábrák készítésénél/kezelésénél is figyelni kell az összes grafikus formátum problémára:

- ▶ pixel grafika (bitmap) (MEGFELELŐ FELBONTÁS),
- ▶ vektor grafika,
- ▶ ha az ábrán vannak betűk akkor annak fontjai (bitmap/vektor),
- ▶ ábrák mérete (bounding box).

Érdemi ábra részek kivágása (bounding box)

- ▶ EPS: ghostview
- ▶ PDF: Acrobat professional (fizetős),  
Briss <http://briss.sourceforge.net/> (ingyenes)

# Ábrák beillesztése

A `figure` környezet szintén egy úszó objektum, melybe grafikus fájlokat lehet beilleszteni. A beillesztés az `\includegraphics{filename}` utasítással történik és a `graphics` csomagot szükséges betölteni a használatához. A `\label{cimke}`, `\caption{felirat}` ugyanúgy használható, de itt csak az ábra alá lehet írni.

Ha az ábra nem megfelelő szögben áll, vagy nem megfelelő a mérete, a `\rotatebox{fok}{mit}` és `\resizebox{vízszintes}{függőleges}{mit}` parancsokkal módosíthatunk rajtuk. Az utóbbinál a két paraméter egyikénél használható a felkiáltójel, jelentése: az arányokat tartsd.

# Ábrák beillesztése – példák

Példa a figure környezet alkalmazására:

```
\usepackage{graphics}
...
\begin{figure}[bp]
\center
\resizebox{80mm}{!}{
\rotatebox{-90}{
\includegraphics{maki.eps}}}
\caption{A \LaTeX\ órák mestere}
\label{fig:maki}
\end{figure}
```

# Hivatkozások automatizálása új utasítások létrehozásával

A hivatkozások egyszerű kezeléséhez az alábbi utasítások definiálását javaslom (ezekkel jóval kevesebbet kell gépelni):

```
\newcommand{\secref}[1]{\aref{sec:#1}.~fejezet}
\newcommand{\figref}[1]{\aref{fig:#1}.~ábr}
\newcommand{\tabref}[1]{\aref{tab:#1}.~táblázat}
\newcommand{\Secref}[1]{\Aref{sec:#1}.~fejezet}
\newcommand{\Figref}[1]{\Aref{fig:#1}.~ábr}
\newcommand{\Tabref}[1]{\Aref{tab:#1}.~táblázat}
```

...

`\Figref{maki}`án egy majmot láthatunk.  
Szép `\figref{maki}`a.

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: Táblázatot és ábrát tartalmazó dokumentum készítése ábra valamint táblázat hivatkozásokkal.

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 2 000–5 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje az óra vége.



# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

## Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

**Hatodik gyakorlat**

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

A matek képletek szedése soron belül is fontos, még ha csak egy betűről is van szó. Pl. melyiket könnyebb (gyorsabb) elolvasni az alábbi két sor közül?

Ha  $a$  pozitív, az negatív, akkor  $z$  negatív.

Ha  $a$  pozitív,  $az$  negatív, akkor  $z$  negatív.

Sose felejtsetek el soron belül matematikai módba váltani, ha szükséges, még ha csak egyetlen betűről is van szó!

Ha  $\$a\$$  pozitív,  $\$az\$$  negatív, akkor  $\$z\$$  negatív.

<https://en.wikibooks.org/wiki/LaTeX/Mathematics>

# Matematikai szedés

A  $\LaTeX$  egyik legnagyobb előnye más szövegszerkesztőkhöz képest a szép matematikai szedés. Három matematikai módot különböztetünk meg egymástól:

- ▶ soron belüli matematikai mód ( $\$$  dollárjelek  $\$$ , vagy  $\backslash( \backslash)$  ilyenek közé kerül a matematikai kifejezés, vagy a `math` környezetbe),
- ▶ külön sorba kerülő, de nem számozott egyenlet (`displaymath` környezetbe, vagy  $\backslash[$  ilyenek  $\backslash]$  közé írjuk a matematikai kifejezést),
- ▶ számozott, külön sorba írt egyenlet (`equation` környezet). Az utóbbi címkézhető (`\label{cimke}`-vel) és hivatkozható.

Számos parancsot csak matematikai módban szabad kiadni!

# Példa

Mindegy, hogy  $a^2 + b^2 = c^2$ , vagy  
 $\backslash(a^2 + b^2 = c^2\backslash)$ , vagy

```
\begin{math}
a^2 + b^2 = c^2
\end{math},
```

ugyanazt láthatjuk a kimeneten.

Viszont a

```
\[ x_{1,2} = \frac {-b \pm \sqrt{b^2 - 4ac}} {2a} \], a
$$$ x_{1,2} = \frac {-b \pm \sqrt{b^2 - 4ac}} {2a} $$$,
```

```
\begin{displaymath}
x_{1,2} = \frac {-b \pm \sqrt{b^2 - 4ac}} {2a}
\end{displaymath}
```

is ugyanaz.

## Hivatkozás képletekre

Az equation környezet számozott, tehát címkézhető; hivatkozás formája (12) – zárójelben szerepel, tehát `(\ref{eq:valami})` –, vagy `\eqref{eq:valami}`

Magyar névelővel `\az\eqref{eq:valami}` vagy esetleg `\aref({eq:valami})`

...mindenki tudja, hogy a  $\pi$  közelíthető `\aref({eq:pi})` kifejezéssel. Mégis megadjuk:

```
\begin{equation}
\label{eq:pi}
\pi = \lim_{n \to \infty} \sqrt{\sum_{k=1}^n \frac{6}{k^2}}.
\end{equation}
```

...mindenki tudja, hogy a  $\pi$  közelíthető az (1) kifejezéssel. Mégis megadjuk:

$$\pi = \lim_{n \rightarrow \infty} \sqrt{\sum_{k=1}^n \frac{6}{k^2}}. \quad (1)$$

**Kiemelt képletek esetében az írásjelek (pont, vessző) mindig a környezeten belülre, a formula végére kerülnek.**

# Szóközök matek módban

Matek módban abszolút nem számít, hogy szóközt üt az ember:  $\$ab_{=}a_{=}b\$,$  pl. így néz ki:  $ab = ab$ . Matek módban nem szabad üres sort írni (nincs új bekezdés).

A pont alapértelmezése: tizedespont (amerikai program), a vessző: felsorolás, melynek következtében  $\$3,14\$\$$  így néz ki: 3, 14.

Megoldás: tizedes törteknél negatív térköz ( $\backslash!$ ), mellyel:  $\$3, \backslash!14\$\$$  = 3,14.

A kapcsos zárójelekkel blokkosíthatunk ( $\{ \dots \}$ ), amire gyakran van szükség ( $\$x^y + z \backslashneq x^{\{y + z\}}\$$ ).

Relációjelek. Az  $=$ ,  $<$ ,  $>$  ugyanúgy használható matek módban. A nem egyenlő, kisebb egyenlő és nagyobb egyenlő jelek a  $\backslashne$ ,  $\backslashle$ ,  $\backslashge$  parancsokkal hívhatóak elő.

# Alapműveleti jelek

Összeadás, kivonás, osztás a megszokott +, -, / jelekkel lehetséges. Vegyük észre, hogy matek módban másképp mutatnak ezek a jelek:  $(2+3)/5=3-2$  simán és  $(2 + 3)/5 = 3 - 2$  matek módban.

A szorzásjelet nem \* -gal jelöljük! Mérnöki gyakorlatban a csillag jelentése: konvolúció. Helyette nem írunk semmit, vagy középmagas pontot ( $\cdot$ ), vagy  $\times$ -t ( $\times$ ) teszünk.

Alsó index az  $_{\text{valami}}$ , felső index a  $^{\text{valami}}$  utasításokkal lehetséges. Vigyázat! Egy matematikai elemnek csak egy alsó és egy felső indexe lehet. Indexek egymásba ágyazása blokkosítással lehetséges (lásd  $\pi^{\pi^{\pi}}$ ). Erre is figyelni kell:  $x^{y^z} \neq (x^y)^z$ :  $x^{y^z} \neq (x^y)^z$  (vegyük észre a z-k különbözőségét).

Görög betűk. Szimplán  $\alpha$ ,  $\beta$ ..., nagy betűk nagy kezdőbetűvel, pl.  $\Phi$ ,  $\Omega$ ...

# Gyökjel

Gyökjel az `\sqrt{valami}` utasítással lehetséges. A  $\text{\LaTeX}$  – szebben mint a MSWord – többet is egymásba tud ágyazni:

```
\begin{equation}\label{eq:emeletes}
\lim_{n \to \infty} \left( \sqrt{\frac{1}{\frac{1}{n} + \sqrt{\frac{1}{\frac{1}{n} + \sqrt{\dots}}}}} \right) = ?
\end{equation}
```

$$\lim_{n \rightarrow \infty} \left( \sqrt{\frac{1}{\frac{1}{n} + \sqrt{\frac{1}{\frac{1}{n} + \sqrt{\dots}}}}} \right) = ? \quad (2)$$

Ha a gyökjel kitevőjét meg akarjuk változtatni, akkor az opcionális paraméterrel tehetjük meg: `\sqrt[3]{2}`  $\approx 1,26$ :  
 $\sqrt[3]{2} \approx 1,26$ .



# Törtjel, binomiális együttható, zárójelek

Nagy törtjel a `\frac{számláló}{nevező}` utasítással lehetséges (lásd a korábbi példákon). Kombinatorikai kifejezésekben hasznos lehet az „enalattaká”, amit `\binom{n}{k}` azaz  $\binom{n}{k}$  vagy a `{n \choose k}` utasítással jelenítünk meg. Ha zavar minket a zárójel, akkor lehet `{n \atop k}` azaz  $\frac{n}{k}$  is.

Zárójelek. A hagyományos kerek és szögletes szimplán, a kapcsos `\{, \}` módon. Ha méretre nem stimmel, akkor célszerű használni a `\left\{` és a `\right\}` parancsokat. Jelentése: a `\left` és a `\right` közötti tartalom magasságához igazítsd a zárójelek méretét.

Ha valahova nem akarunk valódi zárójelet (csak a határt akarjuk jelölni), akkor `\left. .-t,` vagy `\right. .-t` kell írni. Figyelem! A `\left` és a `\right` egy sorban kell szerepeljen! (később lesz jelentősége).

# Norma és egyéb delimiterek

Normaérték jele `|`, az abszolútértéké `|`, pl.

`\sqrt{(a - b)^2} = |a - b|`. Ha két vektor (sík) párhuzamosságát akarjuk jelölni, akkor a `\parallel` parancsra van szükség: `e \parallel f`, azaz  $e \parallel f$ .

Ha valami miatt mégse jó a keretezés (amit a `\left` és `\right` utasításokkal csináltunk), akkor létezik négy méretmódosító utasítás: `\big(`, `\Big(`, `\bigg(`, `\Bigg(`, amit tetszőleges nyitó és záró zárójelekre

`(`, `[`, `\{`, `|`, `|`, `\rfloor`, ... alkalmazhatunk.

A méretmódosító utasítások tipikus felhasználása a több soros képletekben történik.

# Műveletek és operátorok

Szumma, produktum és integrál a `\sum`, `\prod`, `\int` utasításokkal lehetséges.

`displaymath` módban az alsó és felső index az operátor alá és fölé kerül.

`math` módban az alsó és felső index az operátor mellé kerül.

Módok közti váltogatás `displaystyle`, `textstyle`.

Deriváláshoz a „d” operátort a `\mathrm{d}` paranccsal állíthatjuk elő. Ha vesszőt használunk, akkor a ’ is tökéletes:

`\$f'(x) = \frac{\mathrm{d} f(x)}{\mathrm{d} x}\$`

$$f'(x) = \frac{df(x)}{dx}.$$

Az integrálásban és deriválásban is szerepel a „d” operátor!

Gyakori használat esetén célszerű létrehozni egy új utasítást:

`\newcommand*\{dd}\{\mathrm{d}\}`.

# Függvények

A  $\LaTeX$  a gyakori függvényekre beépített paranccsal rendelkezik, pl. módon `\arccos`, `\sin`, `\tanh`, `\exp`, `\ln` stb.

Szöveg bevitele a `\textnormal{szöveg}`, vagy a `\textrm{szöveg}` paranccsal lehetséges:

`\[ x \geq 0, \textrm{ ha } \$x\$ \text{ nemnegatív} \]` A `\textnormal` parancs azonban nem követi a környezeti beállításokat ezért az esetek többségében a `\textrm` parancsot érdemes használni.

Kettőspont általában relációjelekben szerepel (pl.  $:=$ ), ezért ha nem relációjelként szeretnénk használni, akkor a `\colon` parancs kell, pl. `\$f\colon x \mapsto x^2\$`, ami  $f: x \mapsto x^2$ .

Pontok halmozása – ugyanúgy, mint a szövegben – a `\ldots` paranccsal lehetséges az alapvonalra. Vigyázat, az angol kultúrájú országokban középre rakják a pontokat, ha műveleteket hagyunk ki vele: `x_1 + \cdots + x_n`. Létezik függőleges (`\vdots`) és ferde (`\ddots`) pontozás is.

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: Matematikai képletek  $\LaTeX$ -ben való tördelése.

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 5 000–10 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje az óra vége.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

## Matematikai képletek I

## Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

**Hetedik gyakorlat**

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

## Képlettömb

Az eqnarray környezet alkalmas igazított egyenletek bevitelére. Gyakorlatilag egy {rc1} típusú tömb, melyben az oszlopokat &-tel, a sorokat \\-pel választjuk el. Csillagos változata számozatlan egyenleteket szűr be, csillag nélküli változatában a \nonumber paranccsal lehet kikapcsolni a számozást egy sorban. Pl.

```
\begin{eqnarray}
1^3 + 2^3 & = & (1 + 2)^2 \nonumber \\
1^3 + 2^3 + 3^3 & = & (1 + 2 + 3)^2 \\
& & \vdots \nonumber \\
\sum_{i=1}^n i^3 & = & \left( \sum_{i=1}^n i \right)^2
\end{eqnarray}
```

$$\begin{aligned}
 1^3 + 2^3 &= (1 + 2)^2 \\
 1^3 + 2^3 + 3^3 &= (1 + 2 + 3)^2 \\
 &\vdots \\
 \sum_{i=1}^n i^3 &= \left( \sum_{i=1}^n i \right)^2
 \end{aligned} \tag{3}$$

## Túlságosan hosszú egyenletek

Ha túl hosszú az egyenletünk, a  $\LaTeX$  nem fogja automatikusan törni, mert nem tudja, hogy hol törheti az egyenletet. Túl hosszú egyenleteknél is az `eqnarray` környezetet kell használni. Hasznos lehet a `\lefteqn{}` parancs, mellyel az argumentum hosszát virtuálisan `0pt`-re állíthatjuk. Ha az egyenlőségjel két oldalán lévő távolság nem tetszik, akkor használjuk a `\setlength\arraycolsep{2pt}` parancsot.

```
\setlength\arraycolsep{2pt} \begin{eqnarray*}
\lefteqn{f(x) = f(x)\Big|_{x=x_0} + (x - x_0)}
f'(x)\Big|_{x=x_0} \ \&\& \ + \frac{(x - x_0)^2}{2}
f''(x)\Big|_{x=x_0} + \dots + \frac{(x - x_0)^n}{n!}
f^{(n)}(x)\Big|_{x=x_0} + \dots, \end{eqnarray*}
```

$$f(x) = f(x)\Big|_{x=x_0} + (x - x_0)f'(x)\Big|_{x=x_0} + \frac{(x - x_0)^2}{2}f''(x)\Big|_{x=x_0} + \dots + \frac{(x - x_0)^n}{n!}f^{(n)}(x)\Big|_{x=x_0} + \dots,$$



# Túlságosan hosszú egyenletek

Vagy írhatjuk úgy is, hogy

```
\begin{eqnarray} f(x) & = & f(x)\Big|_{x=x_0} + \\ & (x - x_0) f'(x)\Big|_{x=x_0} \nonumber + \frac{ \\ \{(x - x_0)^2\} & 2 f''(x)\Big|_{x=x_0} \\ & \\\ & + \dots \\ & + \frac{\{(x - x_0)^n\}}{n!} f^{(n)}(x)\Big|_{x=x_0} + \\ & \dots \\ \end{eqnarray}
```

$$f(x) = f(x)\Big|_{x=x_0} + (x-x_0)f'(x)\Big|_{x=x_0} + \frac{(x-x_0)^2}{2}f''(x)\Big|_{x=x_0} + \dots + \frac{(x-x_0)^n}{n!}f^{(n)}(x)\Big|_{x=x_0} + \dots \quad (4)$$

# Tömbök matematikai módban

Gyakran van rájuk szükség (pl. mátrix ábrázolása), az array környezet biztosítja nekünk. Használata ugyanaz, mint a tabular környezeté, de matek módban csak ez megy.

```
\begin{equation}
|x| = \left\{ \begin{array}{r}
x & \text{ha } x \geq 0, \\
-x & \text{ha } x < 0.
\end{array} \right.
\end{equation}
```

$$|x| = \begin{cases} x & \text{ha } x \geq 0, \\ -x & \text{ha } x < 0. \end{cases} \quad (5)$$

# Tömbök matematikai módban

Az egységmátrixot pedig az

```
\[ \mathbf{I} = \left( \begin{array}{cccc} 1 & & 0 & & \cdots & & 0 \\ 0 & & 1 & & & & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & & 0 & & \cdots & & 1 \end{array} \right) \]
```

képlet írja le.

Az egységmátrixot pedig az

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

képlet írja le.

## Tömbök matematikai módban

De használhatjuk az amsmath csomag mátrix környezetit (`\pmatrix{}`, `\bmatrix{}`, `\Bmatrix{}`, `\vmatrix{}`, `\Vmatrix{}`) is, amelyek alapértelmezésben megfelelő számú `{cccc}` oszlop formázást készítenek. A csillagos változatokban az oszlop formázás változtatható.

```
\dots\ például
\[ \mathbf{I} = \begin{pmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & & 0 \\
\vdots & & \ddots & \vdots \\
0 & 0 & \cdots & 1
\end{pmatrix}. \]
```

... például

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

# Formázás matematikai módban

Emlékezzünk arra, hogy kétféle utasítás létezik: általános érvényű (pl. `\bf`) és argumentumot fogadó (pl. `\textbf`). Az általános érvényű parancsok hatása is korlátozható, ha blokkosítjuk a dokumentumunkat a `{}` jelekkel (csak `{\bf ezt}` vastagon). A továbbiakban a könnyebb érthetőség kedvéért az általános érvényű utasításokat `{\utasítás }` alakban írjuk, míg az argumentumot fogadó utasítások alakja `\utasítás{}` lesz.

Beavatkozás matek módban. A `\mathit{}`, `\mathrm{}`, `\mathbf{}`, `\mathsf{}`, `\mathtt{}`, `\mathcal{}` és `\mathnormal{}` parancsokkal lehet betűtípust váltani matek módban. A leggyakrabban a `\mathbf{}`-re van szükség, mivel ezzel lehet kövér betűt írni (vektorok és márixok esetében lehet fontos). Pl.  $\mathbf{y} = \mathbf{H} \mathbf{z} \cdot \mathbf{x}$ , ami  $\mathbf{y} = \mathbf{H} \mathbf{z} \cdot \mathbf{x}$ .

Görög kisbetűk vastagítására a `\mathbf{}` nem működik, helyette  $\mathbf{\beta}$  `\boldsymbol{\beta}`, ami  $\beta \rightarrow \boldsymbol{\beta}$ .

# Betűtípusok matek módban

További betűtípust módosító parancsok (zárójelben a szükséges csomag): `\mathscr{}` (`[mathscr]{eucal}`), `\mathfrac{}` (`eufrak`), `\mathds{}` (`dsfont`) és `\mathbb{}` (`amssymb`). Tessék őket kipróbálni!

A matematikai képletek „kivastagítása” a `\{\boldmath }` utasítással lehetséges, de ezt text módban kell kiadni! Például Pithagoras mondá:

`\{\bf\boldmath Kedvenc tétel az  $a^2 + b^2 = c^2$ \}`.

Matematikai méretek módosítása az alábbi parancsok egyikével lehetséges: `\{\displaystyle }`, `\{\textstyle }`, `\{\scriptstyle }` és `\{\scriptscriptstyle }`. A parancs kiadása után az őt követő szövegre a környezet végéig hatással lesz a parancs.

Példa:  $a_{\text{kicsi index}}^{\text{normál index}}$  azaz

`\$a_{\scriptscriptstyle kicsi index}^{\text{normál index}}\$`

Ezek a parancsok a szedést (pl. indexek elhelyezése) is módosítják.

# Térközök matematikai képletekben

Gyakran kell „odébb tenni” a magyarázó szöveget, illetve a feleslegesen összefolyó elemek „szétválasztásakor” is szükség lehet rájuk. A már ismert `\`, `(\)` mellett használható a `\:` `(\)` és a `\;` `(\)`. A szintén ismert `\_` utasítás szóközt generál, a `\quad` `(\_)` és `\qquad` `(\_\_\_)` nagyobb térközök beszúrására alkalmas. A `\quad` az aktuális betűtípus „M” betűjének szélessége. A `\!` utasítással lehet negatív térközt beilleszteni (`-\frac{3}{18}\quad, (\)`).

A `\phantom{valami}` utasítással a `valami` nem jelenik meg, hanem csak a helyét hagyja ki a  $\LaTeX$ . Tipikus alkalmazása az alábbi:

$$\${}^{12}_\phantom{1}6\$, {}^{12}_6C.$$

A `\hspace{hossz}` matek módban is használható, de nem javasolt!!!

# Új operátorok létrehozása

Új matematikai operátor a `\mathop{}` parancssal hozható létre. Ha utána írjuk a `\limits` parancsot, akkor az alsó és felső index alá, illetve felé kerül (ez a default). Ha a `\nolimits` parancsot írjuk utána, akkor alsó és felső indexbe kerülnek az indexek. A gyakran használt operátorokat célszerű `\newcommand`-dal definiálni.

Kétváltozós műveleti jelet a `\mathbin{formula}` parancssal definiálhatunk, új relációjelet a `\mathrel{formula}` hoz létre. Ha szeretnénk valamit írni az – akár létező, akár most definiált – relációjel felé, a `\stackrel{amit}{amire}` parancsra van szükség.

Ha automatizálni akarjuk a matek mód előhívását, akkor az `\ensuremath{arg}` parancsot alkalmazhatjuk. Jelentése: ha matek módban vagy, csak hajtsd végre az `arg`-ot, ha nem, akkor kapcsolj matek módba és utána hajtsd végre az `arg`-ot.



```
\providecommand*\log{\mathop{\log}\nolimits}
\newcommand*\R{\ensuremath{\mathbb{R}}}
\dots
```

Jelölje  $\mathbb{R}$  a valós számok halmazát, legyen továbbá  $a, b \in \mathbb{R}$ , ezzel:

```
\[ \log_2 \left( a \mathbin{\circ} b \right)
\stackrel{!}{\mathrel{\sim}} c \]
```

...Jelölje  $\mathbb{R}$  a valós számok halmazát, legyen továbbá  $a, b \in \mathbb{R}$ , ezzel:

$$\log_2 (a \circ b) \stackrel{!}{\sim} c$$

# Matematikai opciók

A dokumentumok elején a `\documentclass` opcionális paramétereivel meghatározhatjuk a külön sorba írt egyenletek általános képét: a `leqno` opciójával az egyenletek számozása bal oldalra kerül. A `fleqn` opcióval pedig az egyenletek nem középre rendezve, hanem balra zárva jelennek meg. Ha nem tetszik a matematikai képletek behúzásának mértéke, akkor a `\setlength\mathindent{távolság}` paranccsal felülírhatjuk azt.

Mára szinte a standard  $\text{\LaTeX}$  részévé válltak az AMS  $\text{\LaTeX}$  csomagok (AMS = American Math Society), pl. `amsmath`, `amsfonts`, `amstext` stb. Ennek részleteiről külön tárgyat indíthatnánk  $\rightarrow$  érdeklődők utánanézhhetnek a <http://www.ams.org/tex/amslatex.html> címen.

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: Brutális képleteket tartalmazó szöveg tördelése  $\LaTeX$ -hel. Különösen érdekel az órán tanultak (matematikai tömbök, egymás alá rendezett egyenletek) megfelelő alkalmazása.

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 10 000–15 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje az óra vége.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

## Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközök, számlálók

Vízszintes és függőleges térközök

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

[A L<sup>A</sup>T<sub>E</sub>X születése](#)

[L<sup>A</sup>T<sub>E</sub>X kimeneti formátumok](#)

[Vector vs. raszter](#)

## [A L<sup>A</sup>T<sub>E</sub>X használatának alapjai](#)

### [A szöveg begépelése](#)

[A dokumentum struktúrálása](#)

### [Alapvető környezetek](#)

[Programkód beillesztése L<sup>A</sup>T<sub>E</sub>X forrásba](#)

[Felsorolások. Leíró, számozott és számozatlan listák](#)

[Idézetek, versek környezetei](#)

[Irodalomjegyzék és arra hivatkozás](#)

### [Hivatkozások, utasítások, környezetek](#)

[Hivatkozás nem irodalomjegyzék elemre](#)

[Új utasítások és környezetek létrehozása](#)

[Tárgymutató \(Index\)](#)

### [Táblázatok, ábrák](#)

[Táblázatok](#)

[Képek, ábrák](#)

### [Matematikai képletek I](#)

### [Matematikai képletek II](#)

## [Tételek, a szöveg formázása](#)

### [Tételek a szövegben](#)

[A szöveg formázása](#)

### [Térközők, számlálók](#)

[Vízszintes és függőleges térközők](#)

[Számlálók](#)

### [Listák formázás, dok](#)

[Listák formázása](#)

[Dokumentumok felszabdalása, egyéb fájlok csatolása](#)

### [Dobozok](#)

### [Táblázatok formázása, csomagok, osztályok](#)

[Táblázatok formázása](#)

[Úszó objektumok formázása](#)

[Csomagok és osztályok](#)

### [További formai elemek](#)

Dok. szerk. L<sup>A</sup>T<sub>E</sub>X alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

[Nulladik gyakorlat](#)

[Első gyakorlat](#)

[Második gyakorlat](#)

[Harmadik gyakorlat](#)

[Negyedik gyakorlat](#)

[Ötödik gyakorlat](#)

[Hatodik gyakorlat](#)

[Hetedik gyakorlat](#)

[Nyolcadik gyakorlat](#)

[Kilencedik gyakorlat](#)

[Tizedik gyakorlat](#)

[Tizenegyedik gyakorlat](#)

[Tizenkettedik gyakorlat](#)

[Tizenharmadik  
gyakorlat](#)

# Tételek, lemmák

Tételek, lemmák gyakran szerepelnek a szövegben. A tételek kezeléséhez az alábbi dolgokra van szükség:

1. A preambulumban definiálni kell egy új tételkörnyezetet a `\newtheorem{körny_neve}{megjelenítendő_szöveg}` paranccsal. Két helyre illeszthetünk be opcionális paramétert, melyek a számlálók állítását és megjelenítését befolyásolják.
2. A dokumentum törzsében `\begin{körny_neve}` és `\end{körny_neve}` parancsok közé kerül a tétel szövege.

A bizonyításokra standard  $\LaTeX$ -ben nincs külön környezet, de definiálhatunk egyet.

# Példa – a szükséges definíciók

```
% Preambulumba!!!
\newtheorem{ttl}{tétel}
    % tétel beillesztésére szolgáló parancs
\newtheorem{lem}{lemma}
    % lemma beillesztésére szolgáló parancs
\newtheorem{pld}{példa}[subsection]
    % a példák számozása a fejezetszám alapján
\newtheorem{fel}{feladat}[section]
    % a feladatok számozása a chapter száma alapján
\newtheorem{prb}[pld]{probléma}
    % a példák számlálóját használjuk a problémákhoz is
\newenvironment*{proof}{\noindent \emph{Bizonyítás:} }
    {\hfill $\Box$ \par}
    % latexsym csomag legyen betöltve!
```

# Példa – használat

A dokumentumban így használhatjuk a definíciókat:

```
\begin{ttl} Kedvenc tétel \end{ttl}
\begin{proof} Triviális. \end{proof}
\begin{lem} Azért kimondok egy még gyengébb tételt
\end{lem}
\begin{proof} Hogyezeddignemjutottazeszembe\dots
\end{proof}
\begin{pld} Lám milyen szépen öröklí az alfejezetszámot!
\end{pld}
\begin{fel} Lám milyen szépen öröklí a fejezetszámot!
\end{fel}
\begin{prb} Ez meg úgy számozódik, mint a példa!
\end{prb}
```

További lehetőségekért lásd a `theorem`, `ntheorem` és `amsthm` csomagokat.



# Példa – használat (Beamer formátum)

## tétel

*Kedvenc tétel*

*Bizonyítás:* Triviális.

## lemma

*Azért kimondok egy még gyengébb tételt*

*Bizonyítás:* Hogyezeddignemjutottazeszembe...

## példa

*Lám milyen szépen öröklí az alfejezetszámot!*

## feladat

*Lám milyen szépen öröklí a fejezetszámot!*

## probléma

*Ez meg úgy számozódik, mint a példa!*

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

## Tételek, a szöveg formázása

Tételek a szövegben

### A szöveg formázása

### Térközök, számlálók

Vízszintes és függőleges térközök

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

Ha olyan helyre akarunk sortörést, ahova a  $\LaTeX$  magától nem tesz, akkor a `\,`, `\\*`, `\\[hossz]`, vagy `\\*[hossz]` utasítások egyikére van szükség. A csillagos változat nem engedi a lap aljára a sortörést. A `hossz` opcionális paraméterrel megadhatjuk, hogy mekkora helyet hagyjon ki a  $\LaTeX$ .

Léteznek beépített távolsággal sort ugró parancsok is, ezek a `\smallskip`, `\medskip` és `\bigskip`. FIGYELEM! A manuális sortörésre igen ritkán van szükség, csak akkor használjuk, ha tényleg kell!

Egy bekezdésként (egy üres sor után) írt sort a `\noindent` paranccsal tehetünk nem bekezdéssé, illetve az `\indent` parancs teszi a sort bekezdéssé, ha azt nem úgy írtuk. E két parancsra általában új utasítások definiálásakor lehet szükség (lásd a `proof` környezetnél)!

# Sorkizárás, túllógó sorok

Ha fordítás során sok olyan sorral találkozunk, melyeket a  $\LaTeX$  nem tudott eltörni, akkor be kell avatkoznunk. A  $\LaTeX$ -ben rengeteg regiszter befolyásolja a szedés minőségét, a regiszterek értékei pedig állíthatóak (erről később lesz szó). A `\lefthyphenmin` és `\righthyphenmin` változókkal meghatározhatjuk a minimálisan szükséges karakterszámot az elválasztójel előtt, illetve utána. Az alábbi beállítások javasoltak magyar nyelv használata esetén:

```
\setlength\lefthyphenmin{2}
% legalább két karakter a kötőjel előtt
\setlength\righthyphenmin{2}
% legalább két karakter a kötőjel után
\setlength\hyphenpenalty{0}
% nem baj, ha elválasztunk a sor végén
\setlength\uchyph{-1}
% ne válasszon el nagybetűvel kezdődő szavakat
```

# Rakoncátlankodó sorok kezelése

Előfordul azonban, hogy csak körülményes módon szabadulhatunk meg a kilógó soroktól (pl. manuális sortörés).

A `\sloppy` utasítással nagyobb szóközöket is megengedünk a szavak között, mint például a fóliák szövege esetében. A `\fussy` utasítással kapcsolhatunk vissza az alapértelmezett üzemmódra. Azonos módon működnek a `sloppy` és `fussy` környezetek is.

Keskeny helyeken célszerű a tömbös (justified) szedés helyett balra, vagy jobbra (esetleg középre) zárt szedést alkalmazni, mellyel elkerülhetjük a sorok egységnyi szélességbe kényszerítését. A balra zárt szedést a `flushleft` környezet, vagy a `\raggedright` parancs aktivizálja. A szöveg jobbra zárásához a `flushright` környezetet, vagy a `\raggedleft` parancsot kell használni. Középre a `center` környezettel, vagy a `\center` utasítással lehet zárni a sorokat. Ha csak eseti jelleggel kell, a környezetek használatára van szükség.

# Szedés, lapdobás, fejezetek elhelyezése

A szedés típusa a `\documentclass` parancs opcionális paraméterével állítható. Egyoldalas szedéshez a `oneside` opciót, kétoldalas szedéshez a `twoside` opciót kell beírni. Az utóbbi esetben a lapok megjelenése különböző lesz a bal és jobb oldalon.

Ha kétoldalas a szedés, és `book`, vagy `report` osztályban dolgozunk, a `\documentclass` parancs opcionális paraméterével állítható, hogy új `chapter` csak jobb oldalon (`openright`), vagy tetszőleges helyen (`openany`) kezdődhet.

A `\clearpage` utasítással lehet lapot dobni, jelentése: kezdj egy új oldalt. A `\cleardoublepage` is új oldalt kezd, de úgy, hogy az jobb oldali lap legyen.

# Többhasábos szedés

A `\documentclass` opcionális paraméterében állítható. A `onecolumn` (alapértelmezett) opció egyhasábos művet állít elő, a `twocolumn` kéthasábosat. A kéthasábos műben a `\onecolumn` és a `\twocolumn` [szöveg] utasításokkal válthatunk az egy- és kéthasábos üzemmód között. A szöveg a két hasáb fölé kerülő bevezető szövege.

Ha két hasáb kevés lenne, akkor a `multicol` csomagot töltsük be. A `multicols` környezet argumentuma meghatározza a hasábok számát (tíznél kisebb), illetve itt is megvan a bevezető szöveg lehetősége. Ha sok hasáb van, célszerű a `\raggedright`-tal balra igazítani. Példa:

```
\usepackage{multicol}
...
\begin{multicols}{3}[Egye fene, három oszlopba szedem
szövegemet:] Hogyha nékem sok pénzem lesz...
\end{multicols}
```

# Betűtípusok változtatása

A dokumentum szintjén történik (a nyomdászok szerint: egy betűcsaláddal kell szedni az egész művet, de a betűcsalád lehetséges alakjait is használni kell). Az Adobe által ingyenesen elérhetővé tett fontok a megfelelő csomag betöltésével használhatóak (pl. times, avant, avantgar, bookman, chancery, charter, courier stb., lásd:

```
/usr/share/texmf/tex/latex/psnfss/*.sty, vagy  
c:\texlive\2016\texmf-dist\tex\latex\psnfss/*.sty).
```

Betűméret megváltoztatása. A `\documentclass` opcionális paraméterében megadható az alapértelmezett betű mérete. Lehetséges értékek 10pt, 11pt és 12pt (alapértelmezés: 10pt).

Médium (papír) megváltoztatása. Szintén opcionális paraméterrel állítható, lehetséges értékei: letterpaper, legalpaper, executivepaper, a4paper, a5paper és b5paper (alapértelmezés: letter, ezért írtuk be az a4paper-t mindig). A landscape opció elforgatja a lapot 90°-kal.



# Betűk alakja

Három jellemző: alak, vastagság, és család.

Alakjukat tekintve a betűk lehetnek állók (`\textup{}`), döntöttek (`\textsl{}`, vagy `{\sl }`), kurzívák, vagy dőltek (`\textit{}`, vagy `{\it }`) és kiskapitálisok (`\textsc{}`, vagy `{\sc }`)

Vastagságukat tekintve lehetnek normál (`\textmd{}`, vagy `{\md }`), illetve félkövér (`\textbf{}`, vagy `{\bf }`) típusúak.

Végül családjukat tekintve lehetnek antikva (`\textrm{}`, vagy `{\rm }`), groteszk (`\textsf{}`, vagy `{\sf }`) és írógép (`\texttt{}`, vagy `{\tt }`) családba tartozók.

A betűk alakját, vastagságát és családját változtató parancsok egymásba ágyazhatók.

# Betűk mérete

Ha szövegen belül változtatni akarunk a betűk méretén a `{\tiny }`, `{\scriptsize }`, `{\footnotesize }`, `{\small }`, `{\normalsize }`, `{\large }`, `{\Large }`, `{\LARGE }`, `{\huge }` és `{\Huge }` parancsokat használhatjuk.

Sorközök távolsága a `doubleSPACE` csomag `spacing` környezetével változtatható, ahogy az alábbi példa mutatja:

```
\usepackage{doubleSPACE}
...
\begin{spacing}{1.5}
...másfeles sorközű szöveg...
\end{spacing}
```

vagy a `\baselineskip` változó értékének állításával. Pl. `\setlength{\baselineskip}{2.1em}`, de ez már a következő gyakorlat anyaga.

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő:

Nagy házi feladat leírása: <https://moodle.hit.bme.hu>

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 25 000–30 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje is a feladat leírásban található.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

### Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

#### Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Hosszméretek

$\LaTeX$ -ben létezik rugalmatlan és rugalmas (stretching) hosszméret. Az előbbit egyszerűen szám mértékegység alakban (egybeírva!) kell megadni. A használható mértékegységek a 187. főlíán láthatóak.

A rugalmas méreteket szám1 [me1] plus szám2 [me2], vagy szám1 [me1] plus szám2 [me2] minus szám3 [me3] alakban lehet megadni, ahol [me $x$ ] egy mértékegység. Az előbbi esetben a hossz szám1 [me1] és szám1 [me1]+szám2 [me2] közötti tetszőleges értéket vehet fel, az utóbbi esetben az intervallum határai szám1 [me1]+szám2 [me2] és szám1 [me1]-szám3 [me3].

Például: `\setlength{\parskip}{10pt plus 5pt minus 3pt}` 7pt és 15 pt közé állítja a parskip értékét az optimális sortörés függvényében.

# Hosszméretek beállítása

A `\setlength{\hp}{hossz}` paranccsal állíthatjuk be egy már létező `\hp` hosszúságparancs méretét. A `\addtolength{\hp}{hossz}` utasítással `hossz` mértékben növelhetjük `\hp` méretét. A `\the\hp` utasítás írja ki a `\hp` aktuális értékét. A `\newlength{\hp}` hoz létre egy új `\hp` hosszúságot.

A `\settowidth{\hp}{szöveg}` a szöveg szélességére állítja `\hp` értékét, a `\settoheight{\hp}{szöveg}` utasítás szöveg alapvonalától – a sor alapvonalától – mért távolságot adja vissza, míg a `\settodepth{\hp}{szöveg}` az alapvonalától mért mélységre állítja `\hp` értékét.

A `\hspace{hossz}` és `\vspace{hossz}` parancsokkal vízszintes és függőleges térközt lehet kihagyni. Csillagos változataik sor-, illetve oldaltörés esetén is kihagyják a megfelelő méretű helyet.

# Hosszméretek beállítása

Példa:

```
\newlength{\gyalu} A „gyulai gyalu” szélessége  
\settowidth{\gyalu}{gyulai gyalu} \the\gyalu\  
(ide „\hspace*{\gyalu}” pont beférne), magassága  
\settoheight{\gyalu}{gyulai gyalu} \the\gyalu,  
mélysége pedig \settodepth{\gyalu}{gyulai gyalu}  
\the\gyalu.
```

A „gyulai gyalu” szélessége 49.54988pt (ide „ ” pont beférne), magassága 6.85999pt, mélysége pedig 2.26498pt.



# Végtelenül rugalmas térközök

A `\vfill` és a `\hfill` utasításokkal illeszthetünk be vízszintes és függőleges végtelenül rugalmas térközt a dokumentumunkba. Ha például külön sorba írjuk, hogy `bal széle \hfill jobb széle`, akkor ezt láthatjuk:

`bal széle` `jobb széle`

hiszen a `\hfill` rugalmasan kitöltötte a teret. A `\hfill` parancs ekvivalens a `\hspace{0pt plus 1fill}` parancssal.

Léteznek még `\hrulefill` és `\dotfill` parancsok is, melyekkel végtelenül rugalmas vízszintes vonalat, illetve pontsorozatot lehet beilleszteni. Aláírandó dokumentumokat ezért célszerű ilyesformán befejezni:

```
\vfill Budapest, \today \hfill \dotfill
```

Budapest, 2024. február 14.

.....

# Több végtelenül rugalmas térköz aránya

Ha több `\hfill`, vagy `\vfill` követi egymást, akkor azok száma aránylik a távolságokhoz. Ha nem akarunk sokat ismételni, akkor a `\stretch{szám}` mértéket is alkalmazhatjuk a `\hspace{}`, vagy `\vspace{}` argumentumaként:

```
eleje \hspace{\stretch{4}} közepe \hspace{\stretch{1}}
vége \\ ugyanaz, mint az \\
eleje \hfill\hfill\hfill\hfill közepe \hfill vége \\
és az \\ eleje \hspace{0pt plus 4fill} közepe
\hspace{0pt plus 1fill} vége \\ ugye?
```

# Mértékegységek összehasonlító táblázata

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté följái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

jele	jelentése	A méret nagysága más egységben							
		pt	pc	dd	cc	mm	in	bp	sp
pt	pont	1	0,0833	0,9346	0,07788	0,3515	0,0138	0,996	65536
pc	pica	12	1	11,215	0,93457	4,2175	0,166	11,955	786432
dd	Didót pont	1,07	0,0892	1	0,08333	0,3761	0,01481	1,066	70124
cc	cicero	12,84	1,07	12	1	4,5128	0,17767	12,792	841489
mm	milliméter	2,845	0,2371	2,6591	0,22159	1	0,03937	2,835	186468
in	hüvelyk	72,27	6,0225	67,5415	5,62846	25,4	1	72	4736287
bp	nagy pont	1,004	0,0836	0,938	0,07817	0,3528	0,01389	1	65782
cm	centiméter	28,45	2,371	26,591	2,2159	10	0,3937	28,346	1864680
sp	skálázott pont (1/65536 pt)								
ex	az aktuálisan használt betűtípusban „x” magassága, most éppen 2.2pt								
em	az aktuálisan használt betűtípusban „M” szélessége, most éppen 4.4pt								
mu	matematikai mértékegység, csak matek módban létezik! (1/18 em)								
fill	végtelenül rugalmas mértékegység								

# Saját címoldal létrehozása

Címoldalt a `titlepage` környezettel definiálhatunk. A címoldal tartalmát a  $\LaTeX$  azonnal megjeleníti, az oldalszámlálót pedig `l`-re állítja. A címoldal környezetben akár több oldalnyi információt is megjeleníthetünk. Példa:

```
\begin{titlepage} { \center
\resizebox{11cm}{!}{\includegraphics{bme.pdf}} \\\[3mm]
{\large \bf Budapesti Műszaki és Gazdaságtudományi
Egyetem} \\\ \vfill {\Large Valami jópofa cím kell
ide} \\\[20pt] {\Large Nomeg egy szerző} \vfill {\bf
Konzulens}: név \vfill {\Large Budapest, 2017.}
\clearpage \mbox{} \vfill \noindent {\large
Mégegyszer a szerző: cím}} \\\[10pt] Copyright
\copyright 2017. All rights reserved ... \\\[12pt]
This document was typeset in \LaTeXe. \vfill
\end{titlepage}
```

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

### Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Általánosságban a számlálókról

A számlálók egész számok, melyeknek nincs mértékegysége. Sok beépített számlálót használtunk eddig is – bár nem vettük őket észre –, pl. a `section` számláló tárolja a fejezetszámot, a `table` számláló pedig a legutóbbi táblázat sorszámát.

A `\newcounter{szml}[ős]` paranccsal hozhatunk létre egy `szml` számlálót, melynek őse `ős` lesz. Ha az `ős` értékét léptetjük (lásd később), az `szml` kinullázódik. A `\setcounter{szml}{szám}` utasítással írhatjuk felül egy számláló értékét. Az `\addtocounter{szml}{szám}` segítségével egy tetszőleges szám-ot adhatunk `szml`-hez. A `\theszml` kiírja `szml` aktuális értékét, értékátadáshoz pedig a `\value{szml}` utasítást kell használni. Példa:

```
\newcounter{szaml}
\setcounter{szaml}{\value{page}}
Az aktuális oldalszám \theszaml
```

Az aktuális oldalszám 190.

# Beépített számlálók a $\text{\LaTeX} 2_{\epsilon}$ -ben

neve	értéke
part	rész sorszáma
chapter	fejezet sorszáma
section	szakasz sorszáma
subsection	alszakasz sorszáma
subsubsection	al-alszakasz sorszáma
paragraph	paragrafus sorszáma
subparagraph	alparagrafus sorszáma
page	oldalszám
equation	egyenlet sorszáma
figure	ábra sorszáma
table	táblázat sorszáma
numi	lista sorszáma az első szinten
numii	lista sorszáma a második szinten
numiii	lista sorszáma a harmadik szinten
numiv	lista sorszáma a negyedik szinten
footnote	lábjegyzet sorszáma
secnumdepth	fejezet számozás „mélysége”
tocdepth	a tartalomjegyzék „mélysége”

# Számlálók léptetése, hivatkozás a számlálóértékekre

Számlálót a `\stepcounter{szml}`, vagy a `\refstepcounter{szml}` parancsokkal lehet léptetni. Az utóbbi annyival tesz többet, hogy a legközelebbi `\label` parancs `szml` értékét fogja hozzárendelni a címkéhez.

A `\theszml` utasítás automatikusan jön létre, ha definiáljuk a `szml` számlálót. A számlálók értékét azonban többféle módon is kiírathatjuk. Ha nem tetszik a kapott `\theszml` utasítás, akkor át kell definiálni.

```
\newcounter{szaml}  
\renewcommand*{\theszaml}{\Roman{szaml}}
```



# Számlálók megjelenítési lehetőségei $\text{\LaTeX} 2_{\epsilon}$ -ben

<code>\arabic{szml}</code>	arab számokkal	1, 2, 3...
<code>\roman{szml}</code>	kisbetűs római számokkal	i, ii, iii...
<code>\Roman{szml}</code>	nagybetűs római számokkal	I, II, III...
<code>\Alph{szml}</code>	nagybetűkkel	A, B, C...
<code>\alph{szml}</code>	kisbetűkkel	a, b, c...
<code>\fnsymbol{szml}</code>	lábjegyzet szimbólumokkal	*, †, ‡...

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: Egy tetszőleges szövegben a számlálók állítása és megfelelő megjelenítése.

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám ~5 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje az óra vége.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

**Listák formázása**

**Dokumentumok felszabdalása, egyéb fájlok csatolása**

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

**Tizedik gyakorlat**

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

#### Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

**Tizedik gyakorlat**

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

## Számozatlan felsorolások formázása

(itemize környezet) Négy mélységűek lehetnek. Az első szinten `\labelitemi`, a másodikon `\labelitemii`, a harmadikon `\labelitemiii`, a negyediken `\labelitemiv` adja meg a használt felsorolási jel kinézetét. Mindegyikük a `\renewcommand` paranccsal átdefiniálható. Az `\item` opcionális paraméterével pedig egyetlen helyen módosíthatjuk azt.

```
\usepackage{pifont}
...
\renewcommand{\labelitemi}{\ding{49}}
\begin{itemize}
\item Jópofa ez a töltőtoll a sor elején\dots
\item[$\circ$] ide most üres karika kellett,
\item {[]de akkor sem kell zavarba esni, ha
,,[' az első karakter]. \end{itemize}
```

Figyelem, a Beamer átdefiniálja a felsorolás elemeket (ezért nem látható a példák eredménye ebben a dokumentumban)!!

(A pifont csomag ding karakterei.)

# Leíró listák formázása

(description környezet) Hat mélységűek lehetnek. Az `\item[szöveg]` parancs segítségével a szöveg címkével jelenik meg egy elem. Ha más kinézetre törekszünk a `\descriptionlabel` paranccsal módosíthatjuk a címkék megjelenését, de ne felejtjük el a szükséges térközt kihagyni (`\labelsep`)! Példa:

```
\begin{description}
\item[Rock'n'roll] a 60-as években kialakult zenei
irányzat. Meghatározó képviselői közé tartozik...
\end{description}

\renewcommand*{\descriptionlabel}[1]
{\hspace{\labelsep} \sc #1:}

\begin{description}
\item[Rock'n'roll] a 60-as években kialakult zenei
irányzat. Meghatározó képviselői közé tartozik...
\end{description}
```

# Számozott listák formázása

(enumerate környezet) Négy mélységig ágyazhatóak egymásba. A négy szintre – a felsorolásokhoz hasonlóan – i, ii, iii, iv ragokkal hivatkozunk. enumi az első szint számlálója, a már ismert módon, \theenumi utasítással írathatjuk ki értékét. A \labelenumi parancs mondja meg, hogy hogyan kell megjelennie a számlálónak a címkében. Példa:

Azt mondják a magyar nyelvű művek szedésekor antikva szöveg elé kurzív betűket kell írni felsorolásban.

Hát legyen:

```
\renewcommand*{\theenumii}{\alph{enumii}}
\renewcommand*{\labelenumii}{\it \theenumii}
\begin{enumerate} \item Az első szinten még arab
számokat láthatunk, \begin{enumerate} \item de a
másodikon már megjelennek a dőlt betűk.
\item A nyomdászok biztosan jobban tudják, hogy miért
\end{enumerate} \item szebb így. \end{enumerate}
```

# Hivatkozás számozott lista elemére – a hivatkozás formájának módosítása

A `\p@enumi` a második szintű listaelemre történő hivatkozás esetén a hivatkozás alakját állítja be. Az „@” jelet tartalmazó parancsok ún. belső  $\text{\LaTeX}$  parancsok, melyek átdefiniálásához egy speciális üzemmódba kell váltani, amely a `\makeatletter` és `\makeatother` parancsokkal lehetséges. Példa:

```
\renewcommand*{\theenumi}{\arabic{enumi}}
\renewcommand*{\labelenumi}{\theenumi.\S}
\renewcommand*{\theenumii}{\roman{enumii}}
\renewcommand*{\labelenumii}{\theenumi--\theenumii.}
\makeatletter \renewcommand*{\p@enumii}{\theenumi.\S--}
\makeatother \begin{enumerate} \item Első paragrafus
\begin{enumerate} \item Első paragrafus első bekezdése
\item Első paragrafus második bekezdése \label{fontos}
\end{enumerate} \item második paragrafus.
\end{enumerate} Lásd \ref{fontos}. Jó, mi?
```



## Automatikusan sorszámozott felsorolások

Az `enumerate` csomag segítségével a L<sup>A</sup>T<sub>E</sub>X kitalálja, hogy hogyan is akarjuk szervezni a listákat. az `1`, `i`, `I`, `a`, `A` karaktereket keresi az `enumerate` környezet opcionális argumentumában: ha talál, akkor a számozás a talált értéknek megfelelően `arabic`, `roman`, `Roman`, `alph`, vagy `Alpha` lesz. Ha több ilyen is van, akkor a nem kívánatosakat kapcsos zárójelbe kell rakni, hogy elkerüljük a félreértést. Példa:

```
\usepackage{enumerate}
...
\begin{enumerate}
[Kovácsék {az} I.~emeleten {laknak},]
\item zavarja őket a másodikon lakó.
\item zavarja őket a harmadikon lakó.
\item zavarja őket a negyediken lakó.
\end{enumerate}
```

Kovácsék az I. emeleten laknak, zavarja őket a másodikon lakó.

Kovácsék az II. emeleten laknak, zavarja őket a harmadikon lakó.

Kovácsék az III. emeleten laknak, zavarja őket a negyediken lakó.

# Általános lista környezet

LaTeX alatt a list környezetből származtatják az eddig megismert listákat, de még a korábban tanult quote, quotation és verse környezeteknek is a list az alapja. Két paramétere van, az első az alapértelmezett címke, a második a beállításokhoz használható.

Példa:

```
%    define "Lcount" as a counter
\newcounter{Lcount}
%    set the "default" label to print counter as a Roman numeral
\begin{list}{Item-\Roman{Lcount}}
%    inform the list command to use this counter
{\usecounter{Lcount}
%    set rightmargin equal to leftmargin
\setlength{\rightmargin}{\leftmargin}}
%    we can now begin the "items"
\item This is the first item
\item And this is the second item
\end{list}
```

Item-I This is the first item

Item-II And this is the second item

# Általános lista környezet

Egy másik példa:

```
\newenvironment*{idezet}
  {\begin{list}}{
  {\setlength{\rightmargin}{\leftmargin}}%
  \item[] , ,\ignorespaces} {\unskip''\end{list}}
...
```

Egy magas beosztású személy a következőt állította munkahelyéről: \begin{idezet}  
Mi itt a hivatalban mindenhez értünk, csak a gazdálkodáshoz és a szervezéshez nem.  
\end{idezet} No comment.

Egy magas beosztású személy a következőt állította munkahelyéről:

„Mi itt a hivatalban mindenhez értünk, csak a gazdálkodáshoz és a szervezéshez nem.”

No comment.

# Egy másik példa

```
\newenvironment*{Bevásárlólista}
{\begin{list}{\$\Rightarrow$ \hfill}
  % ha rövidebb lenne, mint a hely
{\setlength{\rightmargin}{0pt}
 \setlength{\leftmargin}{\parindent}
 \setlength{\itemindent}{\parindent}
  % igazodva a környezethez
 \setlength{\listparindent}{\parindent}
 \setlength{\labelwidth}{.5\parindent}
 \setlength{\labelsep}{.5\parindent}
 \setlength{\topsep}{1ex plus 0.5ex}
  % az x méretéhez igazodva
 \setlength{\partopsep}{0pt}
 \setlength{\parsep}{0.5ex plus 0.25ex minus 0.25ex}
 \setlength{\itemsep}{0.2ex plus 0.1ex minus 0.1ex}}
{\item[] A bontott csirkét el ne felejtse!\end{list}}
```

# Egy másik példa

```
\begin{Bevásárlólista}
\item tej
\item teljes kiörlésű tönkölybúzából készült kenyér a
Lipóti pékség szerdai sütéséből tönkölybúzából készült
kenyér a Lipóti pékség szerdai sütéséből
\item alma
\end{Bevásárlólista}
```

⇒ tej

⇒ teljes kiörlésű tönkölybúzából készült kenyér a Lipóti pékség szerdai sütéséből tönkölybúzából készült kenyér a Lipóti pékség szerdai sütéséből

⇒ alma

A bontott csirkét el ne felejtse!

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

### Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Hosszabb művek felszabdalása

Ha nagy művet írunk, áttekinthetőbb a dokumentum, ha külön fájlokban tároljuk az egyes fejezeteket. A dokumentumok felszabdalását az `\include{}` parancs támogatja.

Argumentumában a fájl nevét kell megadni, akár a `.tex` kiterjesztés nélkül. Az `\includeonly{}` paranccsal megmondhatjuk a  $\text{\LaTeX}$ -nek, hogy mely fájlokat akarjuk valóban betölteni (akkor hasznos, ha egy fájlban dolgozunk és a többi fordítására nem akarunk időt pazarolni).

Az `\include{}` parancs hatására új oldal kezdődik (mivel alapvetően `chapter`-ök beillesztésére tervezték). Ha ez nem tetszik, akkor az `\input{}` felhasználásával oldaltörés nélkül is beilleszthetünk fájlokat.

```
\documentclass[a4paper,10pt]{report} ...
\begin{document} ... \includeonly{bevezeto}
    % most csak a bevezetőn dolgozunk
\include{bevezeto} \include{modell.tex} ...
\include{befejezes} ... \end{document}
```

# Egyéb fájlok csatolása a $\LaTeX$ forráshoz

Több fájlt is lehet egyetlen  $\LaTeX$  forrásba fűzni. A `\documentclass` utasítás előtt a `filecontents`, környezet jelöli a csatolt fájlt. Egyetlen argumentuma a fájl nevét tartalmazza, a környezetbe pedig a fájl tartalmát kell írni. Akár több fájlt is csatolhatunk a módszerrel.

```
\begin{filecontents}{maki.eps}
%!PS-Adobe-2.0 EPSF-1.2
%%Title: Maki majom ...
\end{filecontents}
\begin{filecontents}{magyar.ldf}
\ProvidesFile{magyar.ldf}
  [1996/12/23 v1.3h Magyar support from ...] ...
\end{filecontents}
\documentclass[a4paper,10pt]{report} ...
```



# Interaktivitás

A  $\LaTeX$ -ben létezik egy `\typein[\prncs]{szöveg}` parancs, mellyel a terminálról olvashatunk be. Egyetlen kötelező argumentumában a terminálra küldött szöveget adhatjuk meg. Ha nem adjuk meg az opcionális paraméterét, akkor a begépelte szöveg az `\includeonly` paraméterét határozza meg. Ellenkező esetben az opcionális paraméterként megadott `\prncs` parancsot deklaráljuk. Hasonlóképpen a `\typeout{szöveg}` utasítással írhatunk a terminálra.

...

```
\typein[\elado]{Kérem az eladó nevét!}  
\typein[\vevo]{Kérem a vevő nevét!}  
\center Adásvételi szerződés \\[1cm]  
Mely köttetett \today\ egyrészeől {\it \elado},  
mint Eladó -- továbbiakban Eladó -- másrészeől  
\it \vevo}, mint Vevő -- továbbiakban Vevő --  
között ...
```

# Néhány dolog amire figyeljete a nagy HF-ben!

- ▶ dolláros matek külön sorba írva (`\$ . $` jellegű)
- ▶ frenchspacing nem alkalmazása, és pont utáni mondatvégek
- ▶ description környezet nem alkalmazása (`\item \textbf{.}` jellegű)
- ▶ gondolatjel, hosszú kötőjel nem ismerete (`_` mintázat)
- ▶ nincsenek meg a kért dolgok
- ▶ idézőjel hibásan (" = hüvelykjel)
- ▶ képletek végén nincs írásjel
- ▶ nem úsztatott táblák (`table` környezet elhagyása)
- ▶ tanult parancsok/paraméterek nem ismerete (pl. `eqnarray` környezet minden sorában `\nonumber` parancs)
- ▶ felesleges `\\` újsor utasítások

# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: Egy listák számozásához és a lista elemekre hivatkozáshoz kapcsolódó feladat megoldása.

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 5 00–3 000 legyen.

A házi feladatokat a `http://moodle.hit.bme.hu` oldalon kell beadni.

A feladat leadásának határideje az óra vége.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

**Tizenegyedik gyakorlat**

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Dobozokról általában

A dobozok segítségével a dokumentumban bizonyos szövegrészeket becsomagolhatunk.

A dobozok fajtái:

**LR doboz** (Left-to-Right) balról jobbra töltjük fel tartalommal, tipikusan csak egy sor

**Bekezdésdoboz** Valamilyen szélességgel bíró, akár több sort is tartalmazó doboz

**Vonaldoboz** Kitöltött téglalapok rajzolására használjuk, illetve gyámfázásra

**Tárolódoboz** Segítségével el tudunk tárolni szövegrészeket (lásd a verses példákat)

# LR dobozok

Ha egy dobozban balról jobbra egyetlen sort írhatunk, akkor LR (Left to Right) doboznak hívjuk. Már korábban is volt szerencsénk a legegyszerűbb dobozhoz, amelyet az `\mbox{szöveg}` utasítással hívhatunk elő. Az `\fbox{szöveg}` hatása szintén doboz, ám ekkor egy keret is megjelenik a `szöveg` körül.

A keretezett dobozban az `\fboxrule` tárolja a keret vonalvastagságát, az `\fboxsep` pedig a keret távolságát a szövegtől. Példa:

```
{\setlength{\fboxrule}{3pt}\fbox{vastag keretes doboz}}
```

**vastag keretes doboz**

```
{\setlength{\fboxsep}{0pt}  
\fbox{térközmentes keretes doboz}}
```

térközmentes keretes doboz

## Még több LR doboz

Egyel bonyolultabb lehetőséget kínál az `\makebox [szél] [poz] {szöveg}` és a `\framebox [szél] [poz] {szöveg}` utasítás, ahol `szél` a doboz szélessége. A szélesség megadható tetszőleges hosszúságmértékben, vagy használható a `\width`, `\height`, `\depth`, vagy `\totalheight` mértékek, melyek a szöveg szélességét, magasságát, mélységét és teljes magasságát (magasság + mélység) adják vissza.

A `poz` paraméterrel a szöveg pozícióját adhatjuk meg a dobozban. Értéke lehet `l` (left), `r` (right), vagy `s` (stretch–shrunken). Az alábbi példában egy keretezett dobozt láthatunk, melynek szélessége a beírt szöveg szélességének duplája, és a szöveg a doboz közepére kerül.

A keretezetlen `\makebox [2\width] [s]{\hfill ABC\hfill}` verzió persze csak akkor látszik, ha van körülötte szöveg. `\framebox [2\width] [s]{\hfill}{ABC\hfill}`

A keretezetlen `ABC` verzió persze csak akkor látszik, ha van körülötte szöveg. ABC

# LR dobozok extra funkcióval

A `\raisebox{emel}[magasság][mélység]{szöveg}` az `emel` paraméter értékével (lehet negatív is!) emeli meg a szöveget. Az opcionális paraméterekkel állíthatjuk, hogy mennyi legyen a virtuális magassága, illetve mélysége a doboznak. Akkor lehet erre szükség, ha nem szeretnénk a sorok távolságát változtatni.

```
Ide kerül az \mbox{első} sor. \\
Ide kerül a \raisebox{2ex}{második} sor. \\
Ide kerül a \raisebox{2ex}[0pt]{harmadik} sor.
```

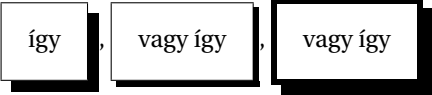
```
Ide kerül az első sor.
                második
Ide kerül a   harmadiksor.
Ide kerül a   sor.
```



# LR dobozok extra funkcióval

A `\shabox{szöveg}` parancs a `shadow` csomag betöltésével hívható elő. Hatására egy árnyékolt keretes doboz jelenik meg. Stílusparamétereit az `\sboxrule` vonalvastagság, az `\sboxsep` térköz és az `\sdim` árnyékvastagság átállításával módosíthatjuk.

```
Jópofa, hogy \shabox{így},  
{\setlength{\sdim}{2pt}\shabox{vagy így}},  
{\setlength{\sboxrule}{2pt}\shabox{vagy így}}  
csinálhatunk árnyékot.
```

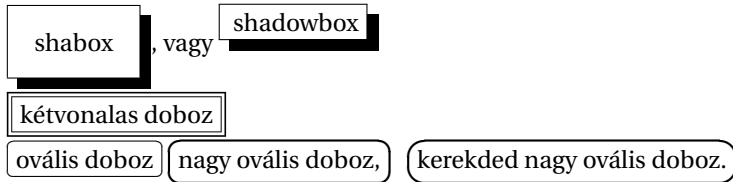
Jópofa, hogy  csinálhatunk árnyékot.

# További extra funkciós LR dobozok

A fancybox csomag további kereteket tartalmaz. A `\shadowbox{szöveg}` hasonló az előbb megismert árnyékolt dobozhoz, de itt az árnyék alja kerül egy vonalba az alapvonallal. Az `\ovalbox{szöveg}` és `\Ovalbox{szöveg}` lekerekített sarkú dobozok, különböző vastagsággal. A `\cornersize{r}` paranccsal változtathatunk a lekerekítés sugarán. A `\doublebox{szöveg}` dupla keretet tesz a szöveg köré, ahol a belső keret vastagsága  $3/4\text{fboxrule}$ , a külsőé  $3/2\text{fboxrule}$ . A fancybox csomagnak további parancsai is vannak, lásd a dokumentációját.

# További extra funkciós LR dobozok

```
\usepackage{shadow}  
\usepackage{fancybox}  
...  
\shabox{shabox}, vagy \shadowbox{ shabox}\  
\doublebox{kétvonalas doboz}\  
\ovalbox{ovális doboz} \  
\Ovalbox{nagy ovális doboz,}  
{ \cornersize{1}  
\Ovalbox{kerekded nagy ovális doboz.}}
```



A `\parbox [poz] [mag] [szpoz] {szélesség}{szöveg}` utasítással több bekezdésnyi szöveget is egy külön dobozba helyezhetünk. A `minipage` környezet is ugyanezekkel az argumentumokkal hívható meg, és hatása is azonos, azonban a `minipage` környezetben lehetnek felsorolások, táblázatok, lábjegyzetek, míg az előbbiben nem. A `poz` paraméter határozza meg a bekezdésdoboz igazítását az alapvonalhoz. Lehetséges értékei: első sorhoz `t` (top), vagy utolsó sorhoz `b` (bottom). Ha elmarad, akkor középre igazít. A `mag` a szövegdoboz magassága, a `szpoz` a szöveg pozíciója a dobozban. Lehetséges értékei: lap tetejére `t` (top), lap aljára `b` (bottom), középre `c` (center) és húz szét `s` (stretch–shrunk).

# Bekezdésdobozok

```
\hrulefill\parbox{25mm}{25\,mm széles doboz sok szöveggel.}%  
\hrulefill\parbox[t]{25mm}{ez is 25\,mm széles doboz,  
máshol.}\hrulefill\parbox[b]{25mm}{ez is 25\,mm széles doboz,  
ismét máshol.}\hrulefill
```

25 mm széles do-  
boz sok szöveg-  
gel.

ez is 25 mm szé-  
les doboz, más-  
hol.

ez is 25 mm szé-  
les doboz, ismét  
máshol.

# Bekezdésdobozok

```
\hrulefill\fbbox{\parbox{25mm}{25\,mm széles doboz sok %  
szöveggel.}}\hrulefill
```

```
\fbbox{\parbox[t][40mm][t]{25mm}{ez is 25\,mm széles doboz,  
máshol.}}\hrulefill
```

```
\fbbox{\parbox[t][40mm][c]{25mm}{ez is 25\,mm széles doboz,  
ismét máshol.}}\hrulefill
```

25 mm széles doboz sok szöveggel.

ez is 25 mm széles doboz, máshol.

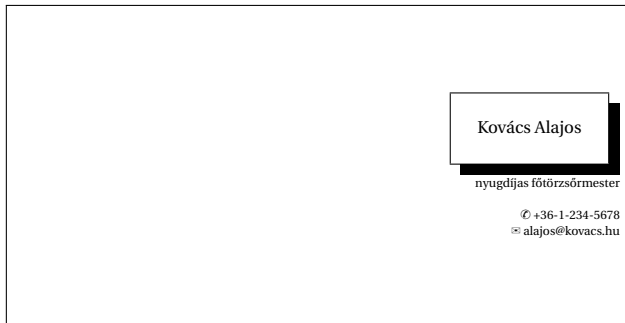
ez is 25 mm széles doboz, ismét máshol.

# Dobozok egymásba ágyazása

Dobozok egymásba is ágyazhatók gond nélkül. Akár nagyobb dobozt rakhatunk egy kisebbbe (néha lehet rá szükség), vagy egyszerűen bármit bekeretezhetünk az `\fbox{}` parancs segítségével. Az alábbi példa egy névjegykártya készítéséhez adhat segítséget.

# Dobozok egymásba ágyazása

```
\usepackage{pifont}
...
\fbbox{\parbox[t][40mm]{80mm}{\vfill \raggedleft
\shabox{\scriptsize Kovács Alajos} \\
\tiny nyugdíjas főtörzsőrmester \\[6pt]
\ding{38} +36-1-234-5678 \\
\ding{41} alajos@kovacs.hu \vfill}}
```





A `\rule[emelés]{szél}{mag}` paranccsal szélszélességű és magmagasságú kitöltött téglalapot rajzolhatunk. Az opcionális paraméterrel megemelhetjük, vagy süllyeszthetjük a téglalapot. A téglalap lehet nulla vastagságú is, amit gyámfának nevezünk. A gyámfázás a nyomdászatban gyakran használt (lásd az első példát). Léteznek beépített gyámfák, melyek a sorvastagság méretét tárolják, ezek a `\strut` szöveg és `\mathstrut` matematikai módban.

# Vonaldobozok

```
\fbox{\rule{0pt}{1cm}ez a doboz legalább 1\,cm magas}\  
\fbox{ez} \fbox{egy} \fbox{egyenetlen} \fbox{dobozsor}  
de  
\fbox{\strut ez} \fbox{\strut már} \fbox{\strut jó} \  
$(\sqrt{g} + \sqrt{h})$ csúnya, de $\left(\sqrt{\mathstrut g} + \sqrt{\mathstrut h}\right)$ szép.
```

ez a doboz legalább 1 cm magas

ez egy egyenetlen dobozsor dex ez már jó

$(\sqrt{g} + \sqrt{h})$  csúnya, de  $(\sqrt{\mathstrut g} + \sqrt{\mathstrut h})$  szép.

Dobozok szövegrészletek tárolására is alkalmasak. A `\newsavebox{\prncs}` utasítással hozhatunk létre egy új, `\prncs` nevű dobozt, melyet háromféleképpen is feltölthetünk. Az `\sbox{\prncs}{szöveg}` utasítást már korábban is használtuk (versidézetek szedésénél), a `\savebox{\prncs}[szél][poz]{szöveg}` annyival több, hogy az opcionális paraméterekkel a doboz szélessége és a dobozban lévő szöveg elhelyezése is befolyásolható. Az `lrbox` környezet szintén az argumentumában megadott `\prncs` doboz feltöltésére alkalmazható (új környezetek definiálásakor lehet hasznos). A feltöltött dobozok tartalmának kiíratása a `\usebox{\prncs}` utasítással történik.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

**Táblázatok formázása**

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föliái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Táblázatok stílusparamétereit

Matematikai módban a `\begin{array}[poz]{oszl}` környezettel, szöveges módban a `\begin{tabular}[poz]{oszl}` környezettel hozhatunk létre tömböket. Az `oszl` paramétert már korábban megismertük, a `poz` paraméter a táblázat illesztésének módját adja meg. Ha `t` (top), akkor a táblázat első sora kerül az alapvonalra, ha `b` (bottom), akkor az alsó sor. Ha nincs megadva ez az opcionális paraméter, akkor a táblázat közepét igazítjuk az alapvonalhoz.

# Táblázatok stílusparamétereit

```
\hrulefill
\begin{tabular}[t]{c} 1 \\ 2 \\ 3 \\ 4 \end{tabular}%
\hrulefill
\begin{tabular}{c} 1 \\ 2 \\ 3 \\ 4 \end{tabular}%
\hrulefill
\begin{tabular}[b]{c} 1 \\ 2 \\ 3 \\ 4 \end{tabular}%
\hrulefill \\
```

			1	
			2	
		1	3	
		2	4	
_____	1	_____	3	_____
	2		4	
	3			
	4			

## Automatikus sortörés táblázatban

Ha megadjuk egy táblázat oszlopdefiníciójában, hogy az oszlop adott szélességű legyen, akkor gyakran csúnya a kimenet, mivel a sorkizárt szedés miatt széles szóközök kerülnek a szavak közé.

Megoldást jelenthet az alábbi kód, melyhez az array csomagot be kell tölteni.

```
\usepackage{array}
...
\newcommand*{\tabsoverage}[1]{\let\tmp=\#1\let\=\tmp}
\begin{tabular}{c|>{\tabsoverage{\raggedright}\mbox{}}
p{25mm}} \hline 1 & Az első sorban is látszik milyen
hasznos kis segítség ez. \\ \hline 2 & Hát még a
második sorban. \\ \hline \end{tabular}
```

1	Az első sorban is látszik milyen hasznos kis segítség ez.
2	Hát még a második sorban.



## Fix szélességű táblázat beillesztése

A `tabularx` csomaggal illeszthetünk be adott szélességű táblázatot. Használatához a `\begin{tabularx}{szél}[poz]{oszl}` környezetet kell meghívni. Az első argumentuma a táblázat szélességét határozza meg. Hasznos lehet a `\linewidth` méret alkalmazása, amely az aktuális vonalszélességet adja vissza. A következő paraméter opcionális; a korábban megismert `poz` pozícionáló paraméterrel ekvivalens. Az utolsó oszlopban az oszlopdeklarációk szerepelnek, azonban itt megjelenik egy speciális oszlop: a rugalmas oszlopot az `X` definíció jelöli.

```
\begin{tabularx}{\linewidth}{c|X} Akkora, amekkora kell
& ez meg a maradékot kapja meg mindenképpen akármit is
írok, legyen az akár nagyon hosszú szöveg \\ Rövidebb
szöveg & második sor. \end{tabularx}
```

Akkora, amekkora kell

ez meg a maradékot kapja meg mindenképpen akármit is írok, legyen az akár nagyon hosszú szöveg

Rövidebb szöveg

második sor.

## Arányok a fix szélességű táblázatban

Ha több X típusú oszlop is van, akkor azok azonos szélességűek lesznek. Ha ez nem tetszik, akkor az oszlopdefiníciókban módosíthatjuk az X típusú oszlopok szélességeinek arányát. Az alábbi példa remélhetőleg megvilágítja a használatát:

```
\begin{tabularx}{\linewidth}
    {c|>{\hspace=0.5\hsize}X|c|>{\hspace=1.5\hsize}X}
1 & megérett a meggy & 2 & csipkebokor vessző \\
3 & te leszel a párom & ... & \end{tabularx}
```

1	megérett a	2	csipkebokor vessző
	meggy		
3	te leszel a	...	
	párom		

Az oszlopdefinícióban szereplő két szám összege meg kell, hogy egyezzen az X oszlopok számával. Jelen esetben az oszlopok szélességének aránya:  $0.5 / 1.5 = 1 / 3$ , azaz a negyedik oszlop szélessége háromszorosa a másodikénak.

# Az oszlopdefiníciók teljes listája

További paraméterek az array csomaggal. A tabular környezet kötelező argumentumában az oszlopdefiníciók kiegészítése:

- `l` balra zárt oszlop
- `r` jobbra zárt oszlop
- `c` középre zárt oszlop
- `p{szél}` sorkizárt oszlop szél szélességgel, az első sor az alapvonalon
- `m{szél}` sorkizárt oszlop szél szélességgel, a sorok közepe az alapvonalon
- `b{szél}` sorkizárt oszlop szél szélességgel, az alsó sor az alapvonalon
- `>{dekl}` az adott oszlop elemei előtt minden sorban a `dekl` deklarációt végrehajtja
- `<{dekl}` az adott oszlop elemei után minden sorban a `dekl` deklarációt végrehajtja
- `|` függőleges vonal
- `@{szöveg}` az oszlopelválasztó felülírása térköz lenyeléssel
- `!{szöveg}` az oszlopelválasztó felülírása térköz lenyelése nélkül
- `*{n}{oszl}` Több ( $n$  darab) azonos oszlopdefinícióval rendelkező oszlop(ok) ismétlése

# Más programok kimenetének gyors táblázatba foglalása

A `dcolumn` csomag betöltése után használható az oszlopdefiníciókban a `D{isep}{osep}{jegy}` paraméter, amely megmondja a  $\LaTeX$ -nek, hogy mivel jelöltük a tizedespontot (`isep`), mivel szeretnénk helyettesíteni (`osep`) és hány digit szerepel az inputban (`jegy`). Az utóbbiba negatív számot írjatok, ha nem akartok bíbelődni.

```
\usepackage{dcolumn}
...
\begin{tabular}{|>{$}r<{$}!{$=$}D{.}{,}{-1}|}
\hline % most jön a Matlab kimenete
1^2 & 1 \\
\sqrt{2} & 1.4142 \\
\sqrt{100} & 10 \\
\hline
\end{tabular}
```

# Hosszú táblázatok

Ha táblázatunk egy oldalnál hosszabb, akkor a `longtable` csomagot kell betöltenünk. A `longtable` környezettel illeszthetünk be egy oldalnál hosszabb táblázatokat. A csomagnak – és a környezetnek – sok opciója van, tessék utánanézni <http://www.cmis.csiro.au/Graham.Williams/TeX/docs/longtable.pdf>.

# Színes táblázatok

A `colortbl` csomaggal a táblázat oszlopai, sorai, cellái  
kiszínezhetőek. Dokumentáció szintén a weben:

<http://www.cmis.csiro.au/Graham.Williams/TeX/docs/colortbl.pdf>

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

**Tizenkettedik gyakorlat**

Tizenharmadik  
gyakorlat

# Vonalak cifrázása

A `\hhline` csomaggal lehetséges a táblázatok dupla vonalait „szépíteni”. Dokumentáció szintén a weben:

<http://www.ctan.org/tex-archive/macros/latex/required/tools/hhline.dtx>

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté főléai alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

**Tizenkettedik gyakorlat**

Tizenharmadik  
gyakorlat

# Táblázatok stílusparaméterei

Az alábbi hosszméreteket lehet átírni a már ismert `\setlength` parancs segítségével. Az `\arraycolsep` és a `\tabcolsep` az oszlopok közötti távolság fele `array`, illetve `tabular` környezet használata esetén. Az `\arrayrulewidth` paranccsal a táblázatban lévő vonalak vastagsága szabályozható. A `\doublerulesep` kettőzött vonalak esetében a vonalak közötti távolság.

A sorok közötti távolságot az `\arraystretch` paraméter határozza meg. Dimenzió nélküli szám, megváltoztatása a `\renewcommand` paranccsal lehetséges. Alapértéke 1.



# Itt ér véget a mai gyakorlat

A hallgatók házi feladata a következő: Ez már a nagy házi feladat! Egy tetszőleges saját, vagy internetről letöltött szöveget kell formázni, amely az alábbi követelményeket teljesíti: van benne folyó szöveg, van benne legalább egy táblázat, van benne legalább egy kép/ábra, van benne legalább egy képlet és tartalmaz irodalomjegyzéket, illetve azokra hivatkozást.

A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 25 000–30 000 legyen.

A házi feladatokat a <http://moodle.hit.bme.hu> oldalon kell beadni.

A feladat leadásának határideje a 12. oktatási hét péntek, reggel 8:00.

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

### Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Úszó objektumok általános formázási szabályai

A `figure` és `table` környezettel hozhatunk létre úszó objektumot. Ha csillagos verzióikat használjuk, akkor kéthasábos szedés esetén a táblázat, vagy az ábra mindkét hasábot elfoglalhatja. Hasznos nagy táblázatok, ábrák beillesztésekor.

A `\caption[rövid]{cím}` utasításnak adhatunk egy rövid opcionális paramétert is, melynek jelentése: a táblázatok, vagy ábrák listájában a rövid címet írd ki. A táblázatok és ábrák jegyzéke a `\listoftables` és a `\listoffigures` utasításokkal íratható ki.

A `wrapfigure` csomag segítségével elérhetjük, hogy minden ábra a beillesztés helyén jelenjen meg.

# Ábra, vagy táblázat körbefolyatása szöveggel

A `floatflt` csomagot kell betölteni, majd a `\begin{floatingfigure}{szél}` környezetbe kell az ábrát illeszteni, pl. a korábbról megismert `\includegraphics{}` utasítással. A `floatingfigure` környezet paraméterében határozhatjuk meg, hogy vízszintesen mekkora helyet foglaljunk az ábrának. Az ábra elé és mögé írt szöveg körbefolyja az ábrát. Hasonlóan használható a `floatingtable` környezet.

```
\usepackage{floatflt}
...
\begin{floatingfigure}{.5\linewidth} \center
\resizebox{.45\linewidth}{!}{\includegraphics{maki.eps}}
\caption{Majom Úr} \label{fig:maki} \end{floatingfigure}
```

A majmok sok tulajdonságukban emlékeztetnek az emberre

# Több részelemből álló ábra

A `subfigure` csomag betöltése után az ábrákon belül alábrákat is létrehozhatunk a `\subfigure [rész cím] {rész ábra}` utasítással. A rész cím csak a rész ábra alatt jelenik meg. Ha van `\caption`, akkor az egész ábrát kitölti.

```
\usepackage{subfigure}
...
\begin{figure} \center \mbox{%
\subfigure[Az állat]{\includegraphics{maki.eps}} \quad%
\subfigure[A ketrec]{\includegraphics{bme.eps}} }
\caption{Beíratnám jó eszű maki egyedemet,
hogya végezze el e Műszaki Egyetemem (kecskerím)}
\end{figure}
```

# Ábrák beillesztésének formázása

Az `\includegraphics` stílusparaméterei. Ha a `graphics` csomag van betöltve, akkor eléggé szegényesen paraméterezhető: két opcionális paraméterében megmondhatjuk, hogy milyen bal alsó és jobb felső koordináták közötti képrészletet akarunk beilleszteni. Ha csillaggal használjuk, akkor csak a képrészlet kerül be a dokumentumba, csillag nélkül a teljes kép (de csak a kivágottnak foglal le helyet)

Ha a `graphicx` csomagot használjuk a `graphics` helyett, akkor sokkal több lehetőségünk van. A dokumentációért lásd a 9–12. oldalakat:

<http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps>

```
Itt jön egy makifej \resizebox{2cm}{!}{\rotatebox{-90}
{\includegraphics*[0,200][400,600]{maki.eps}}}
de most berakja a teljes képet is, csak csacska módon
\includegraphics[0,200][400,600]{maki.eps}
szöveg most például már nem is látszik.
```

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközők, számlálók

Vízszintes és függőleges térközők

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

### Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté fölíái alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Miért és hogyan írhatunk osztályokat és csomagokat?

Csomag írására akkor lehet szükség, ha van néhány utasításunk, melyeket gyakran használunk, függetlenül a dokumentumosztálytól. Saját dokumentumosztályra akkor lehet szükségünk, ha egy új osztályt akarunk definiálni, melyben többen is dolgozni fognak. Az osztály definiálás hosszadalmas és alapos munka, ráadásul sok olyan dolgot is ismerni kellene hozzá, amiről nem esett szó (lásd a példa fejezetcím formázó utasítását). Ezért nem foglalkozunk új osztályok definiálásával, részletekért lásd <http://www.tex.ac.uk/tex-archive/macros/latex/doc/clsguide.ps> és <ftp://cam.ctan.org/tex-archive/macros/latex/base/classes.dtx>.

Csomagok esetében az első sorba a `\ProvidesFile` utasítást kell írni, melynek első paramétere a csomag neve, majd opcionálisan megadhatjuk a csomag készítésének dátumát, verziószámot és szöveges ismertetőt. A csomag használatakor hivatkozhatunk e paraméterekre:

```
\usepackage[csomag opciók]{csomag}[dátum]
```



# Példa csomagra

A jeny.sty file tartalma:

```
\ProvidesFile{jeny.sty}[2002/12/03 v1.0 Probacsomag]
\setcounter{secnumdepth}{5}
    % öt mélységig számozza a fejezeteket
\setcounter{tocdepth}{4}
    % a tartalomjegyzék négy mélységű
\makeatletter
\renewcommand\section{\@startsection {section}{1}{\z@}%
    {-3.5ex \@plus -1ex \@minus -.2ex}%
    {2.3ex \@plus .2ex}{\normalfont\tiny\bfseries}}
\makeatother
\newcommand*\JG{\textsf{Jeny Gábor}}
\newenvironment*{idezet}
    {\begin{list}{}{\setlength{\rightmargin}{\leftmargin}}}%
    \item[,,\ignorespaces] {\unskip''\end{list}}
...
```

# Tartalomjegyzék

## Adminisztratív és egyéb kérdések

A  $\LaTeX$  születése

$\LaTeX$  kimeneti formátumok

Vector vs. raszter

## A $\LaTeX$ használatának alapjai

### A szöveg begépelése

A dokumentum struktúrálása

### Alapvető környezetek

Programkód beillesztése  $\LaTeX$  forrásba

Felsorolások. Leíró, számozott és számozatlan listák

Idézetek, versek környezetei

Irodalomjegyzék és arra hivatkozás

### Hivatkozások, utasítások, környezetek

Hivatkozás nem irodalomjegyzék elemre

Új utasítások és környezetek létrehozása

Tárgymutató (Index)

### Táblázatok, ábrák

Táblázatok

Képek, ábrák

### Matematikai képletek I

### Matematikai képletek II

### Tételek, a szöveg formázása

Tételek a szövegben

A szöveg formázása

### Térközök, számlálók

Vízszintes és függőleges térközök

Számlálók

### Listák formázás, dok

Listák formázása

Dokumentumok felszabdalása, egyéb fájlok csatolása

### Dobozok

### Táblázatok formázása, csomagok, osztályok

Táblázatok formázása

Úszó objektumok formázása

Csomagok és osztályok

### További formai elemek

Dok. szerk.  $\LaTeX$  alatt

Telek Miklós

BME, Hálózati  
Rendszerek és  
Szolgáltatások Tanszék

(Jeney Gábor és Horváth  
Máté föléli alapján)

Nulladik gyakorlat

Első gyakorlat

Második gyakorlat

Harmadik gyakorlat

Negyedik gyakorlat

Ötödik gyakorlat

Hatodik gyakorlat

Hetedik gyakorlat

Nyolcadik gyakorlat

Kilencedik gyakorlat

Tizedik gyakorlat

Tizenegyedik gyakorlat

Tizenkettedik gyakorlat

Tizenharmadik  
gyakorlat

# Lábjegyzetek formázása

A korábban megismert `\footnote{szöveg}` paranccsal lehetséges. Opcionális argumentumával `\footnote[szám]{szöveg}` megadhatjuk a használni kívánt számlálóértéket. Ekkor a beépített `footnote` számláló értéke nem változik. A `\footnote{}` utasítás előtt nem szabad szóközt hagyni, lehet az szó végén, vagy írásjel után. A `footnote` számláló értékét a már ismert `\thefootnote` utasítással lehet kiírni. Ha nem tetszik a lábjegyzet formája, az utóbbit kell felülírni.

```
\renewcommand*{\thefootnote}{\roman{footnote}}
Mindenki\footnote{az első lábjegyzet}
másképp\footnote[20]{hát ez meg hogy lett ilyen nagy?!}
Pedig még csak \thefootnote\ az értéke!!!}
csinálja\footnote{kettes számú lábjegyzet}%
\dots\footnote[1]{Vigyázat, becsapós!}
```

# Lábjegyzetek – külön a jelölés, külön a tartalom

A `\footnotemark [szám]` paranccsal egy lábjegyzetjelet illeszthetünk be a folyó szövegbe. Ha nem használjuk az opcionális argumentumot, akkor a sorban következő jelet teszi ki és lépteti a `footnote` számlálót eggyel. Ha használjuk az opcionális paramétert, akkor számnak megfelelő lábjegyzetjelet teszi ki és nem bántja a `footnote` számlálót

A `\footnotetext [szám] {szöveg}` a lábjegyzetbe írja szám jellel a szöveget, anélkül, hogy a szöveggörnyezetbe jelet tenne. Ha nem adjuk meg az opcionális argumentumot, akkor az aktuális számlálóértéket írja ki, és a számláló értékét nem változtatja.

```
Mindenki\footnotemark\footnotetext{Közös lábjegyzet.}
másképp\footnotemark[\value{footnote}]
csinálja\footnotemark[\value{footnote}]\dots
```

# Ahol a lábjegyzet nem használható direkt módon

A `\footnote{}` parancs nem használható sem dobozokban (minipage környezetben a környezet aljára kerülnek a lábjegyzetek, nem a lap aljára), sem táblázatokban. Ha ilyen környezetekben szeretnénk `\footnote` utasítást normál módon használni, akkor gányolni kell a `\footnotemark` és a `\footnotetext` utasításokkal. Közben persze ügyesen trükközni kell a `footnote` számláló értékével.

```
Ez a megoldás\footnote{Megoldás?!} is \fbox{jó%
\footnotemark[101]},\footnotetext[101]{Csak nem szép.}
mert rosszul mutat\footnote{Nocsak.} az ugrás.
A jó megoldás sem sokkal bonyolultabb, lássuk hát
\fbox{\strut{A-ban\footnotemark} B-ben\footnotemark}!
\addtocounter{footnote}{-1}\footnotetext{A-ban?}
\stepcounter{footnote}\footnotetext{B-ben?}
```

# Lábjegyzetek formai paramétere

Állítható a lábjegyzetek közötti távolság (`\footnotesep`), a lábjegyzetek és a szöveg közötti vonal kinézete (`\footnoterule`), melynek 0 pt vastagnak kell lennie (lásd a lenti példát). A `\skip\footins` méret változtatásával a szöveg és a lábjegyzet első sora közötti távolság állítható be. További stílusmódosítási lehetőségeért lásd a `footnote` csomagot.

```
\setlength{\footnotesep}{1.5cm} % biztosan látjuk majd
\renewcommand*{\footnoterule}{\rule{\linewidth}{.3pt}}
\vspace{-.3pt}} % a vonalvastagságot minuszba is
                % kihagyjuk, hogy össz 0pt legyen
\setlength{\skip\footins}{1cm}
Mindenki\footnote{az első lábjegyzet} másképp
csinálja\footnote{kettes számú lábjegyzet}%
\dots\footnote{hármasszámú lábjegyzet!}
```

# Széljegyzetek

A `\marginpar{szöveg}` utasítással írhatjuk a szöveget a margóra. Alapértelmezésként mindig a kötés átellenes oldalára kerül a szöveg. Ha ez nem tetszik, akkor `\reversemarginpar` utasítással a másik oldalra kényszeríthetjük a margófeliratot. Vissza a `\normalmarginpar` utasítással léphetünk.

A széljegyzetek formai megjelenését a `\marginparwidth`, a `\marginparsep` és a `\marginparpush` paraméterekkel állíthatjuk. Az első a széljegyzetek szélességét szabályozza, a második a széljegyzet és a szedéstükör távolságát, az utolsó két egymást követő széljegyzet közötti minimális függőleges távolságot állítja be.

A `\marginpar[bal szöveg]{jobb szöveg}` utasításnak létezik egy opcionális paramétere is, mellyel a páros oldalakra eltérő szöveget írhatunk (`bal szöveg`), mint a páratlan oldalakra (`jobb szöveg`).

# Élőfej, élőláb

A `\pagestyle{sty}` utasítással a dokumentum egészét, a `\thispagestyle{sty}` utasítással pedig az adott lap stílusát határozhatjuk meg. A `sty` stílus lehet `empty`, (nincs fejléc és lábléc), `plain`, (nincs fejléc, a láblécben középen az oldalszám), `headings`, (a lábléc üres, a fejlécben az oldalszám kívül, belül a rész cím és fejezetcímek).

A `fancyhdr` csomag betöltésével és a `fancy` stílus megadásával a fejléc és lábléc tartalma tetszőlegesen befolyásolható. A `\fancyhead[hely]{mit}` és `\fancyfoot[hely]{mit}` parancsokkal írhatunk a fejlécbe és láblécbe. A `hely` argumentum lehet `L` (left), `R` (right), `C` (center), `E` (even – páros oldalon), `O` (odd – páratlan oldalon). Egy utasításon belül több hely is megadható, ezeket vesszővel kell elválasztani. Az alábbi kód a lapok külső oldalára fejlécbe írja az oldalszámot.

```
\usepackage{fancyhdr} ...  
\pagestyle{fancy} \fancyhead[LE,R0]{\thepage}  
\fancyfoot{}
```



A tartalomjegyzék beillesztése a `\tableofcontents` paranccsal lehetséges, az ábrák jegyzéke a `\listoffigures`, míg a táblázatoké a `\listoftables` utasítással íratható ki.

A jegyzékek automatikusan generálódnak fordítás során, de lehetséges beléjük írni is. Ha szeretnénk formázó utasítást írni az egyik jegyzékbe, akkor az `\addtocontents{hova}{mit}` utasítást kell használnunk, ahol a `hova` paraméter a jegyzék típusa (`toc`, `lof`, vagy `lot`). Ha egy elemet akarunk beilleszteni, akkor az `\addcontentsline{hova}{típus}{mit}` utasítást írjuk, ahol a `típus` mondja meg a beillesztett mező típusát (tartalomjegyzék esetében `part`, `chapter`, `section` stb. lehet, a másik két esetben csak `figure`, illetve `table`). A megjelenő oldalszám az utasítás kiadásának helyét mutatja.

# További olvasmányok – könyvek

A legjobb magyar könyv a témában: Wettl Ferenc, Mayer Gyula, Sudár Csaba, *L<sup>A</sup>T<sub>E</sub>X kezdőknek és haladóknak*, Panem, 1998, Budapest

A legjobb könyv a témában: Leslie Lamport, *L<sup>A</sup>T<sub>E</sub>X; A Document Preparation System* Addison-Wesley, 1994.

Aki a T<sub>E</sub>X rejtelseit is szeretné felfedezni: Donald E. Knuth, *The T<sub>E</sub>Xbook*, Addison-Wesley, 1984.

Aki a T<sub>E</sub>X-et magyarul szeretné felfedezni: Bujdosó Gyöngyi, Fazekas Attila, *T<sub>E</sub>X kezdőlépések*, Tertia, 1997, Budapest

# További források az Interneten

Ha valakinek nincs pénze könyvekre, akkor az Interneten is rengeteg jó forrást fedezhet fel. A „ $\TeX$  frequently asked questions” weboldal:

`http://www.tex.ac.uk/tex-archive/macros/latex/doc/`

$\LaTeX$  navigator (és encyclopedia): `http://tex.loria.fr/`

A  $\TeX$ -hel és  $\LaTeX$ -hel kapcsolatos anyagok gyűjtőhelye (Comprehensive  $\TeX$  Archive Network – CTAN):

`ftp://ftp.dante.de`, vagy

`ftp://helka.iif.hu/mirrors/ctan/`

A  $\LaTeX$  hivatalos honlapja: `http://www.tex.ac.uk/CTAN/latex/`

A magyar  $\LaTeX$  honlap címe: `http://www.math.bme.hu/latex/`

# Konvertálás más formátumokba

A `dvi1j`, `dvi1j4` stb. konverterekkel HP LaserJet alatt nyomtatható formátumot állíthatunk elő a DVI kimenetünkből.

A `dvips` alkalmazással PostScript (PS) kimenetet állíthatunk elő. A UNIX világban a PS a kiindulási alap, PostScriptből bármibe (akár mozgó GIF-be is!) `gs`-sel (GhostScript) alakíthatjuk át fájljainkat. A `gs` dokumentációját a `man gs` paranccsal hívhatjuk elő.

Lehetőségünk van közvetlenül PDF formátumba is konvertálni. Erre a `dvipdf`, alkalmazás szolgál, mely DVI-ből készít PDF-et. A `pdflatex` alkalmazással a  $\LaTeX$  forrásunkat közvetlenül PDF-be fordíthatjuk, de vigyázat, nem a  $\LaTeX$ 3 csoport fejlesztte, ezért a portabilitás nem garantált és lehetnek benne hibák. Ettől függetlenül sok kellemes szolgáltatása van a `pdflatex`-nek, például az `\includegraphics` parancsnak közvetlenül GIF és JPG fájlokat is átadhatunk.

# Egyéb konverterek

Egyéb konverterek is léteznek, melyek nem részei a  $\text{\LaTeX} 2_{\epsilon}$  disztribúciónak. A teljesség igénye nélkül ilyen például az `rtf2latex`, amely RTF fájlokat tud  $\text{\LaTeX}$  forrásba átírni, a `latex2rtf`, amely ugyanezt csinálja csak fordítva, vagy a `latex2html`, amely HTML kimenetet állít elő a  $\text{\LaTeX}$  forrás alapján. Mivel az alkalmazásokat felsorolni is lehetetlen, a webes keresőket ajánlom további információért és bővebb dokumentációért.

<http://www.google.com>

# VÉGE

Ennyi fért a félév anyagába, talán hasznos is volt. Remélem sikerült megismerni a  $\LaTeX$  világát és alkalmatok nyílik később használni az itt tanultakat.

A tárgy anyagával kapcsolatban várom észrevételeiteket (mit csináltál volna másképp, mi hiányzott, mi volt sok stb.) a [telek@hit.bme.hu](mailto:telek@hit.bme.hu) e-mail címre. Ha a későbbiekben lenne kérdésed a  $\LaTeX$ -hel kapcsolatban, akkor is kereshetsz ezen a címen.